

# Instance and semantic segmentation of point clouds of large metallic truss bridges

Daniel Lamas, Andrés Justo, Mario Soilán, Manuel Cabaleiro, Belén Riveiro \*

CINTECX, Universidade de Vigo, GeoTECH Group, Campus Universitario de Vigo, As Lagoas, Marcosende, Vigo 36310, Spain

## ARTICLE INFO

### Keywords:

Point clouds  
Truss bridge  
Panoptic segmentation  
Semantic segmentation  
Instance segmentation  
LiDAR  
TLS

## ABSTRACT

Several methods have been developed for the semantic segmentation of reinforced concrete bridges, however, there is a gap for truss bridges. Therefore, in this study a state-of-the-art methodology for the instance and semantic segmentation of point clouds of truss bridges for modelling purposes is presented, which, to the best of the authors' knowledge, is the first such methodology. This algorithm segments each truss element and classifies them as a chord, diagonal, vertical post, interior lateral brace, bottom lateral brace, or strut. The algorithm consists of a sequence of methods, including principal component analysis or clustering, that analyse each point and its neighbours in the point cloud. Case studies show that by adjusting only six manually measured parameters, the algorithm can automatically segment a truss bridge point cloud.

## 1. Introduction

Transport infrastructure is a vital asset for our societies because it enables the movement of people and the exchange of goods. Bridges are one of the most important infrastructure elements, which are used by several modes of transport. They can overcome obstacles to landscapes such as rivers, canals, and valleys. Bridges have become crucial to the existence or development of certain territories. A notable example of great benefit to the EU is the Oresund Bridge [1], which connects Sweden and Finland with the rest of the EU by road [2].

Unfortunately, a significant number of bridges in transportation networks are aging, causing a global and critical issue. According to the European Commission's Science and Knowledge Service [3], many of the 1234 km of road bridges over 100 m in length in the EU, which were built during the economic boom of the 1950s, have now reached the end of their design life, and carry more vehicle traffic than expected. Because of this concern and because manual inspection is still the most common method to assess the condition of bridges, research into better bridge maintenance techniques is required.

Building information modelling (BIM) [4] is a principal solution for infrastructure maintenance. When applied to bridges, this is called bridge information modelling (BrIM). BrIM technology enables the storage of bridge data, such as 3D models and component specifications, in a digital file. This is extremely useful not only during the design and

construction of new bridges, but also during the maintenance and inspection of in-service bridges. Bouzas et al. [5] presented a methodology based on a heritage-BIM framework to create models with both architectural and structural information, using data obtained from point clouds, accelerometers, and geometric measurements made in situ. The different changes experienced by the bridge, such as maintenance work, can be integrated into the model, and the structural model can be exported to structural analysis software. Shalabi et al. [6] generated a BrIM model, which incorporated inspection data and a 3D representation of an in-service bridge, demonstrating that this technology substantially improves bridge assessment and maintenance operations. Salamak et al. [7] generated not only a BrIM model based on design documentation and a point cloud obtained with laser scanning, but also added augmented reality technologies to improve maintenance tasks.

However, the development of BrIM models may not be an easy task, particularly for replicating bridge geometries in digital models. Thus, several techniques are being developed to digitise the as-is geometry and automate this process. To do so, it is necessary to use innovative solutions, such as remote sensing technologies, that allow massive gathering of geometric data on bridges with an unprecedented level of accuracy. Laser scanning, particularly terrestrial laser scanning (TLS), is one of the most widely used technologies for this task. A vast body of literature has demonstrated these capabilities [8–10]. Rashidi et al. [11] reviewed more than 1500 research publications on the applications of TLS in

\* Corresponding author.

E-mail addresses: [daniel.lamas.novoa@uvigo.gal](mailto:daniel.lamas.novoa@uvigo.gal) (D. Lamas), [andres.justo.dominguez@uvigo.gal](mailto:andres.justo.dominguez@uvigo.gal) (A. Justo), [msoilan@uvigo.gal](mailto:msoilan@uvigo.gal) (M. Soilán), [mcabaleiro@uvigo.gal](mailto:mcabaleiro@uvigo.gal) (M. Cabaleiro), [belenriveiro@uvigo.gal](mailto:belenriveiro@uvigo.gal) (B. Riveiro).

<https://doi.org/10.1016/j.autcon.2023.104865>

Received 28 September 2022; Received in revised form 23 March 2023; Accepted 5 April 2023

Available online 21 April 2023

0926-5805/© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

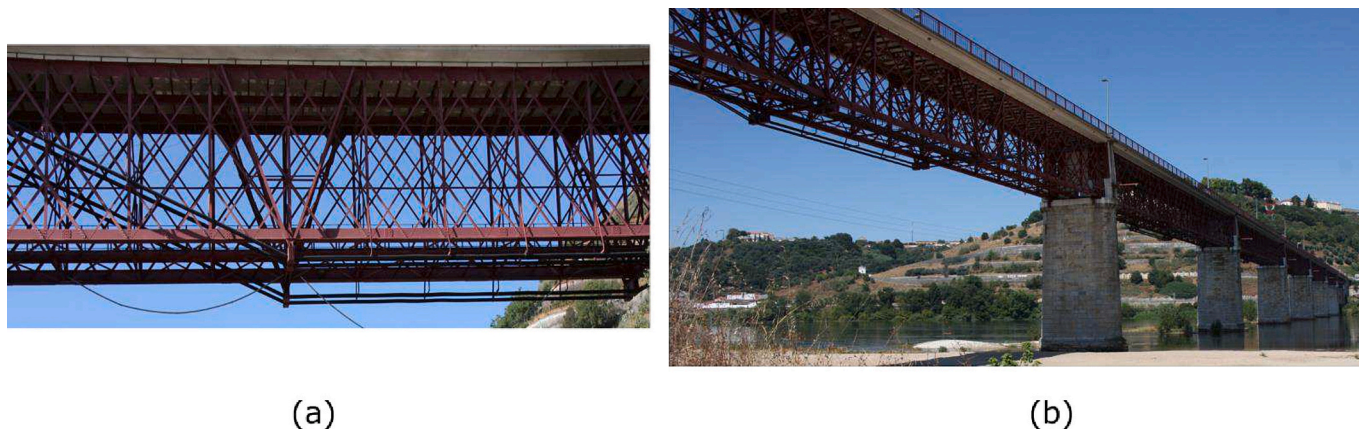


Fig. 1. Case study I.

bridges for generating 3D models, inspection tasks, structural assessments, and BrIM models. Although this technology can generate representations of the environment in the form of 3D point clouds, the generation of a 3D model is not automatic and requires segmentation and classification processes to identify and locate the bridge elements.

Based on this, and the idea of automating these processes, several authors have developed algorithms to automatically segment the point clouds of bridges. Lu et al. [12] presented a slicing method for segmenting structural elements in reinforced concrete (RC) bridge point clouds. Their method can deal with complex topologies and reduce the computational cost by splitting data. Moreover, it can process point clouds with occlusions, local variability density of points, and frequent events in the point clouds. Yan and Hajjar [13] proposed a similar heuristic-based method for segmenting steel-girder bridges based on their geometric and topological constraints. In a previous study, we [14] developed an automated process for detecting the components of masonry bridges based on normal surface calculations. However, these methods cannot be applied to truss bridges, which are more complex than those analysed in previous studies.

Several authors have presented deep-learning methods for segmenting point clouds. PointNet [15] is a state-of-the-art architecture that directly applies a deep-learning method to points for classification. Despite being overtaken by other models, most are based on this model and its improved version, Pointnet++ [16]. Recently, a new model presented by Zhao et al., called the Point Transformer [17], proposed an entirely different architecture based on the work of Jaderberg et al. on spatial transformer networks [18]. For the bridges, Kim et al. [19] presented a PointNet-based neural network for the segmentation of RC bridges. The classes considered were the deck, pier, and background. The consideration of the background significantly reduces the time-consuming pre-processing of the point cloud. Xia et al. [20] presented an automated semantic segmentation method that applied a multiscale local descriptor and used the results to fit a classification artificial neural network. They implemented a local descriptor to compensate for the lack of data required to train the network. Their results were compared with those obtained using Pointnet [15], which was trained with only a few data used in their work, obtaining an improvement in the mean intersection over union from 46% to 95%. Their work confirmed the value of a descriptor when there is a lack of data to train a model. However, this method classifies points by point, obtaining a segmented point cloud, but does not determine which points belong to which element, which is necessary to obtain a 3D model of the bridge. Therefore, Xia et al. [20] refined their results by creating clusters using DBSCAN [21]. However, this algorithm does not work if there are elements of the same type which are intertwined with each other, such as in a truss bridge, because the algorithm forms a unique cluster with these elements. Hu et al. [22] presented a method for reconstructing the

structure-aware semantic 3D models of cable-stayed bridges. They used unmanned aerial vehicle (UAV) images and the corresponding photogrammetric point clouds as inputs. Two independent neural networks extracted features from both types of data. These features are combined and decoded to a hierarchical binary parsing tree using BiTreeNet and to 3D shapes from the shape nodes in the binary parsing tree using their Shape Decoder to obtain the 3D bridge model.

In the field of truss bridges, Gyetvai et al. [23] developed a workflow to create a model for conducting a finite element-based structural assessment of a truss bridge point cloud. Their study can be divided into two distinct processes. First, the process identifies the cross section of each element by comparing the section-based point cloud to a library of sections. Second, the primary dimensions of the bridge are estimated to generate the model. However, the segmentation process for each item was not automatic. Shang et al. [24] presented a method for 3D reconstruction of truss bridges using images instead of point clouds. They proposed flight planning for UAVs to capture bridge images by reducing occlusions. These images were used to construct the bridge meshes. However, their model only contained information about the bridge as an entity, and not about each of its elements.

This study aims to contribute to current methods for the automatic generation of 3D truss bridge models from point clouds. To the best of our knowledge, there is no automatic process for segmenting truss bridge point clouds. Therefore, the aim was to develop an algorithm capable of automatically segmenting the point clouds of truss bridges, from which sufficient information can be extracted to create a geometric model of the structure containing the nodes, where each element is connected. This study starts with the hypothesis that by knowing the location of a point of each bridge item and its main direction, it is possible to define the nodes of the structure. Hence, the main objective of this study is to develop automatic instance segmentation to at least partially detect each item whose main direction matches the main direction of the item to which it belongs.

This paper is structured as follows. First, case studies are presented in section 2. Section 3 details the proposed methodology for the segmentation of a 3D point cloud of a truss bridge. Section 4 presents the results, which are then discussed in section 5. Finally, section 6 presents the conclusions of this study. Repository available at [https://github.com/GeoTechUVigo/truss\\_bridge\\_pointcloud\\_segmentation](https://github.com/GeoTechUVigo/truss_bridge_pointcloud_segmentation).

## 2. Case studies

This section presents case studies used to validate the methodology.

### 2.1. Case study I

The first case study is a truss beam bridge, shown in Fig. 1(a) and

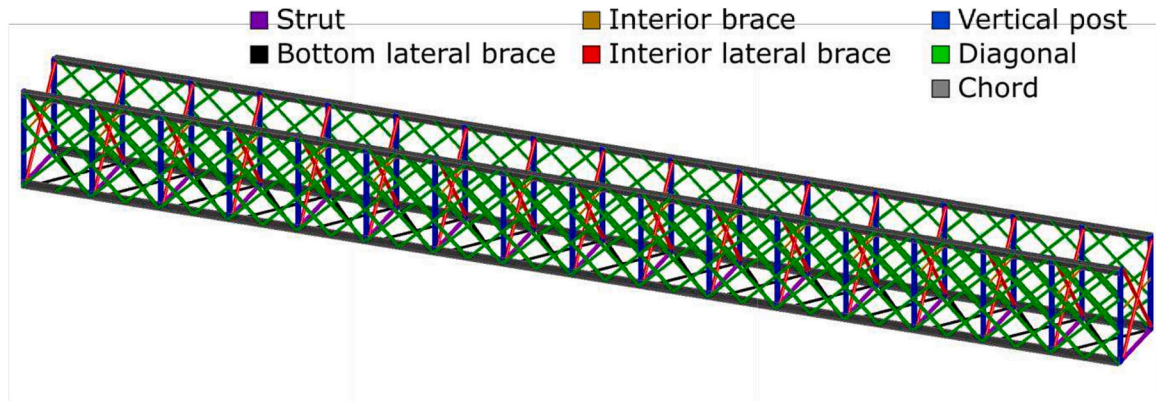


Fig. 2. Representation of case study I.

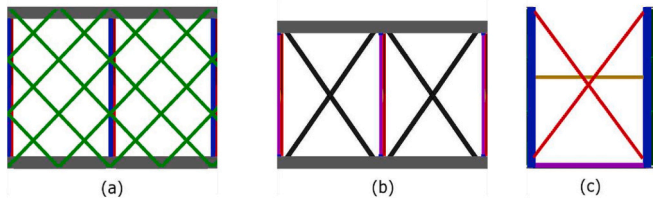


Fig. 3. Case study I faces. (a) Vertical face, (b) horizontal face, and (c) interior face.

Fig. 1(b).

The truss structure between two pillars of the bridge was repeated for each pair of pillars. Fig. 2 shows a representation of the structure between the two pillars. The truss can be divided into vertical, horizontal, and interior faces. The two vertical faces shown in Fig. 3(a) contain vertical posts and diagonals. The horizontal face represented in Fig. 3(b), which is the bottom face, comprises struts and bottom lateral braces that form Saint Andrew’s Crosses separated by struts. The 17 interior faces, represented in Fig. 3(c), each comprise a Saint Andrew’s Cross formed by two interior lateral braces traversed by an interior brace. In addition, four chords joined and supported the entire truss.

2.2. Case study II

The second case study was 114 m long and divided into two distinct parts with different structures. The analysed part is the truss beam structure, which is 30 m long and rests on two pillars. The truss structure can be appreciated in Fig. 4(a) and Fig. 4(b).

A schematic of the bridge truss structure is shown in Fig. 5. The two vertical faces shown in Fig. 6(a) contain vertical posts and diagonals. The horizontal or bottom face, as shown in Fig. 6(b), comprised struts and bottom lateral braces. These lateral braces form Saint Andrew’s

Crosses separated by struts. The interior faces, which are represented in Fig. 6(c) contain interior lateral braces and vertical posts. In addition, four chords join and support the entire truss, and two stringers support the bridge deck.

3. Methodology

The methodology covers data collection during the segmentation process, including the identification of infrastructure components, locations, and dimensions. As shown in Fig. 7, the proposed methodology is a sequential procedure that can be divided into three steps.

The first step, data acquisition, involves the acquisition of bridge data in the form of a 3D point cloud. Section 3.1 explains the field operations conducted to obtain the data used to validate the methodology.

The second step, pre-processing, prepares the data to be analysed. This is discussed in detail in section 3.2.

The third processing step was segmentation. The segmentation methods allow the identification of each member of the bridge. First, the bridge is split into faces, as explained in section 3.3. Each type of face is then analysed, as explained in section 3.4. The analysis of the vertical, horizontal, and interior faces is explained in sections 3.4.1, 3.4.2 and 3.4.3., respectively.

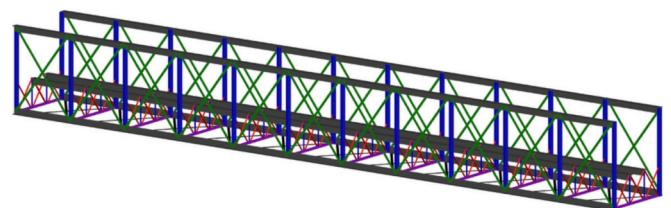


Fig. 5. Representation of case study II. Colour legend is in Fig. 2.

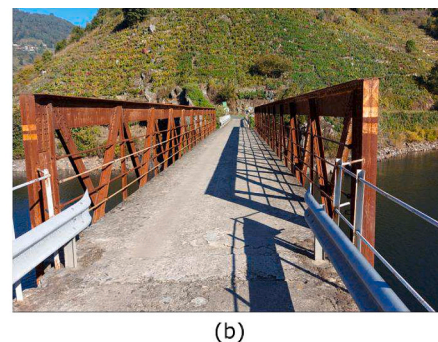
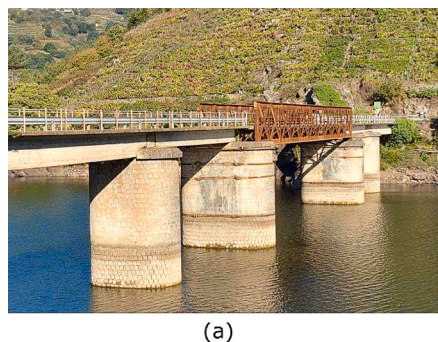


Fig. 4. Case study II.



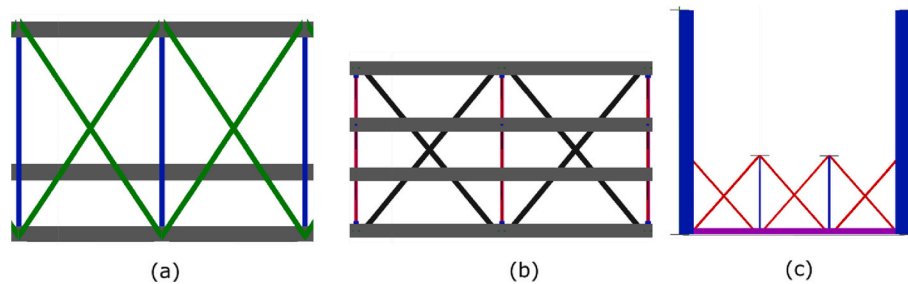


Fig. 6. Case study II bridge faces. (a) Vertical face, (b) horizontal face, and (c) interior face.

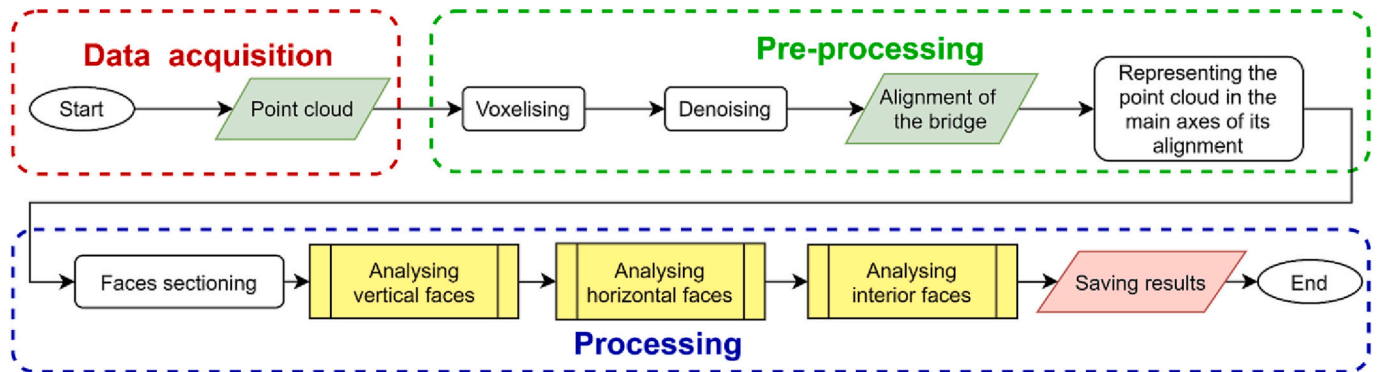


Fig. 7. Proposed methodology workflow.

Throughout this process, different sub-processes are executed at different points in the methodology. For instance, the method used to segment the diagonals of the lateral faces is also used to segment the bottom lateral braces of the horizontal faces and the interior lateral braces of the interior faces. Section 3.4 explains how each face is analysed and section 3.5 details the truss element segmentation process.

Workflow diagrams are presented to explain the algorithms. All diagrams have two ellipses representing the start and end. The input is denoted by a green rhomboid and the output by a red rhomboid. The face analysis processes are represented by the yellow rectangles (explained in section 3.4), orange truss element segmentation processes (explained in section 3.5), blue specific processes, and white single processes.

Section 3.1 explains the data acquisition. Section 3.2 explains the pre-processing. Section 3.3 explains the face sectioning. Section 3.4 explains the vertical, horizontal, and interior faces analysis. Section 3.5 explains the methods used in the face analysis to segment each type of item of a truss bridge.

### 3.1. Data acquisition

The case studies were surveyed using a TLS (Faro Focus X330). This instrument is a phase-shift laser scanner with an operating range of 0.6–120 m. The Focus X330 has a nominal precision of  $\pm 2$  mm at 25 m distance in normal reflectivity and illumination conditions. This instrument has a maximum angular resolution of  $0.009^\circ$  a horizontal field of view of  $360^\circ$  and a vertical field of view of  $305^\circ$ . The acquisition rate of the instrument reaches up to 1 million points per second when working at maximum speed.

#### 3.1.1. Case study I

Restricted access to the case study I structure forced the design of an optimised network of scanner stations from both sides of the river. This network was formed using eight different scanner positions at ground level under the bridge, with 37 scans, to ensure that the whole bridge structure was recorded. The data collection from each scanner position

comprised a low-resolution scan to draft a panorama view to identify the placement of the structure to be scanned in detail. It also included a high-resolution scan of the field of view corresponding to the bridge.

Considering the large size of the bridge to be scanned, a topographic survey was performed to accurately measure targets in the bridge and surroundings that could be used to 1) align all the scans into the same coordinate system, 2) ensure the levelling of the point cloud, and 3) georeference the consolidated point cloud into the Universal Transvers Mercator global coordinate system using a Leica TS15 total station. The technical features of the instrument are as follows: long-range measurements up to 1000 m reflectorless (R1000); 1 mm + 1.5 ppm accuracy; and angular precision of  $1''$  for both horizontal and vertical directions.

A total of 527 points were measured for the entire bridge to allow the alignment of the current scans, as well as aligning the scans measured in future acquisition campaigns. Only a small number of control points were used for the registration of the current scans.

For the bridge span used to explain the processing method presented in the forthcoming sections, eight scanner position data were used, which also ensured the registration of all the scans collected in the south margin of the bridge. These scans were aligned using the control points measured by the total station to obtain a consolidated point cloud with more than 100 million points (more than 20 million points representing the span under analysis), with a final registration error of 1.1 cm. The spatial resolution of the point cloud was between 5 and 15 cm, depending on the distance between the target object and scanner. It is important to mention that, owing to the extreme geometric complexity of the truss structure and limited access to the structure, the consolidated point cloud has important occlusions as well as other sources of errors, such as mixed pixels provoked by the sharp form of the beam elements.

In addition, alignment, which is defined as the central axis of the road crossing the bridge, is used to determine the orientation of the bridge more precisely. The alignment is represented as a polyline formed by XYZ points stored in a TXT file.

Fig. 8 shows the recorded point cloud, and Fig. 9 shows the scan



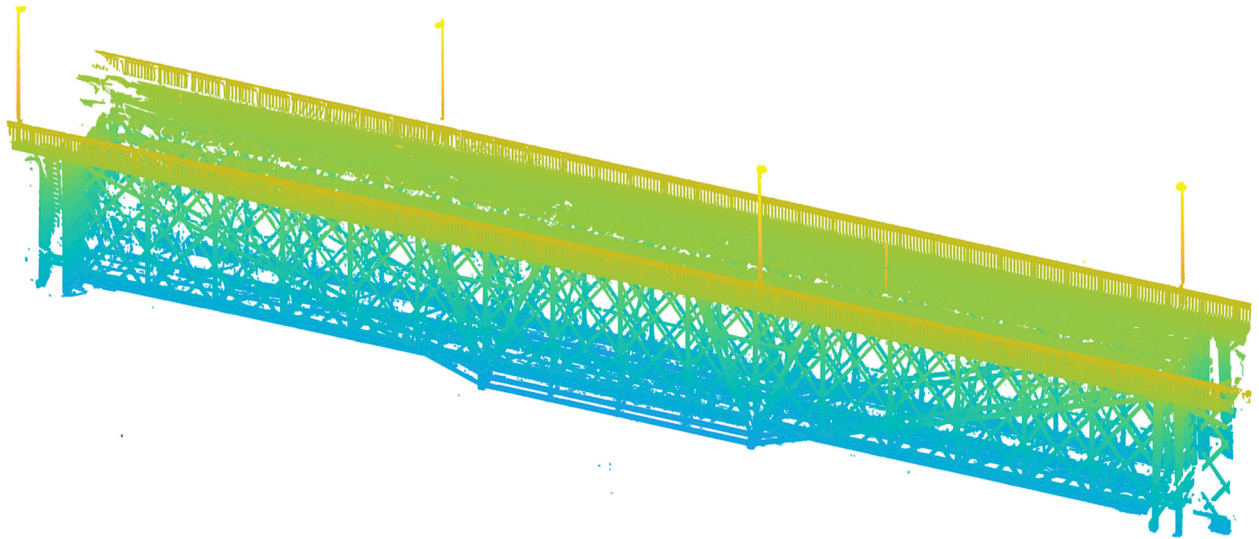


Fig. 8. Raw case study I bridge point cloud.

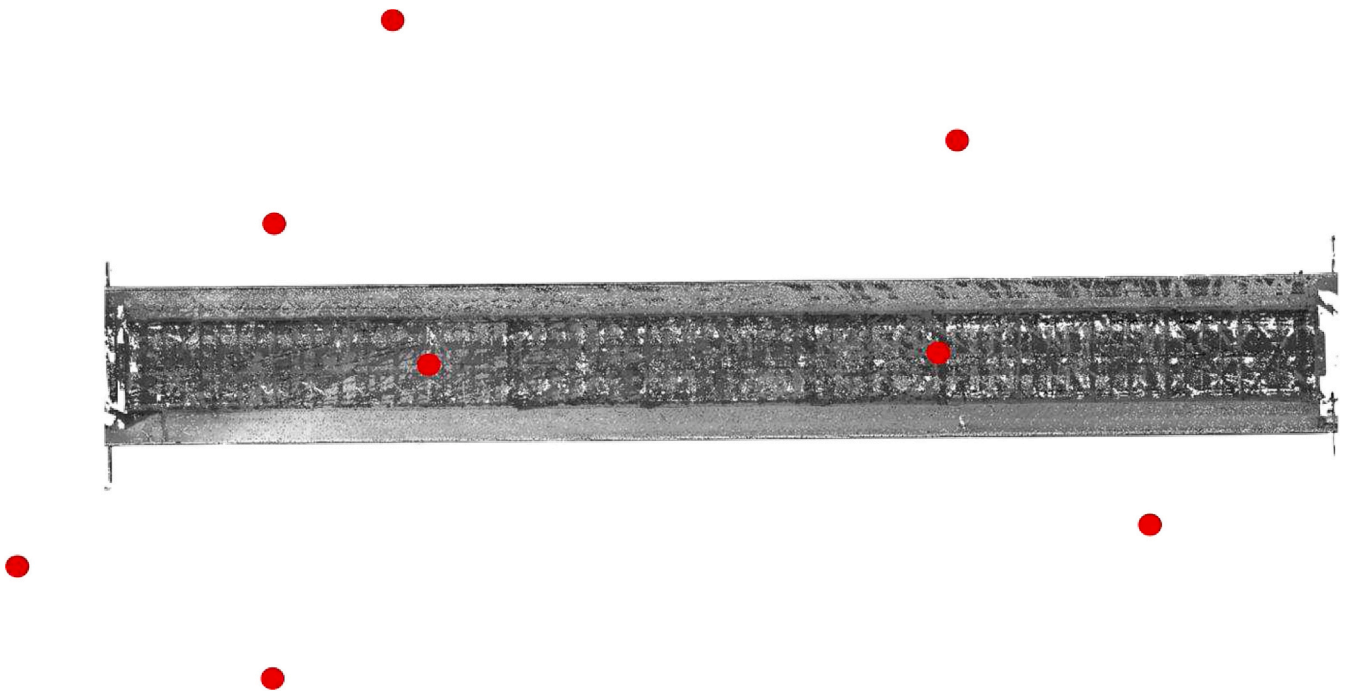


Fig. 9. Eight scan positions of case study I shown as red dots at ground level under the bridge. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

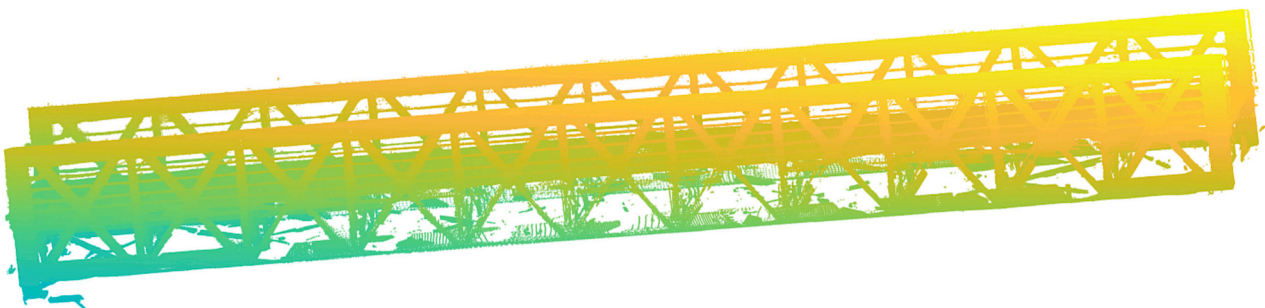


Fig. 10. Raw case study II bridge point cloud.

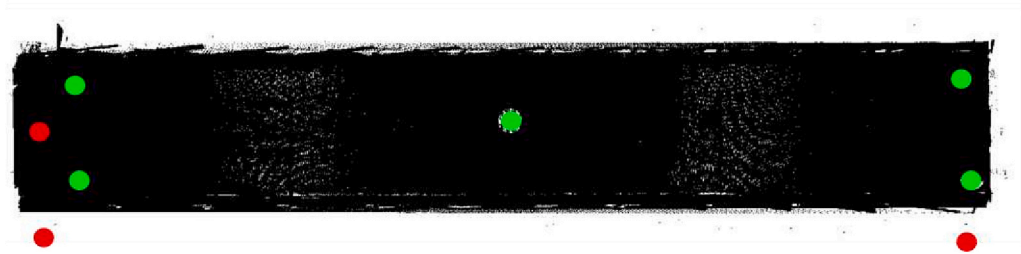


Fig. 11. Nine scan positions of case study II shown as green dots on the bridge, red dots on the pillars, and blue dot is a remote position. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

positions.

### 3.1.2. Case study II

Data acquisition for case study II followed a similar process. Nine scans were performed at different positions, on the deck, pillars, and remote areas on the banks of the dam. Consequently, a 130 million points point cloud of the truss structure was obtained. Figs. 10 and 11 show the recorded point clouds and Figure 11 scan positions, respectively.

### 3.2. Pre-processing

The data obtained from field operations were raw point clouds. These raw point clouds do not have a uniform density. This can hinder the segmentation process because certain operations such as defining objects based on the number of points cannot be applied. Besides, and related to the above, there are regions with more points than necessary,

which increases the computation cost when the point cloud is segmented.

Therefore, a voxelisation process was applied. Chen et al. define a voxelised object as a “3D discrete representation of a continuous object on a regular grid of voxels” [25]. This process consists of discretising the space in a regular grid of cubic cells called voxels. The grid size used in this study is denoted by  $g$ . Several metrics can be applied to the points in the voxels that allow the organisation of point clouds. In this case, the metric used was the calculation of the centre of mass of the point in each voxel. The centres of mass represent the 3D position of each voxel. In this way, voxelisation not only makes the point density uniform but also down samples the point cloud, making it easier for it to be analysed if the level of down sampling is appropriate for the intended purpose. Voxelisation is widely used in the classification and segmentation of point cloud tasks. For instance, some scholars, such as Wang et al. [26], have developed technologies based on the voxelisation of point clouds, to analyse forest structures. In addition, Riano et al. [27] and Weishampel

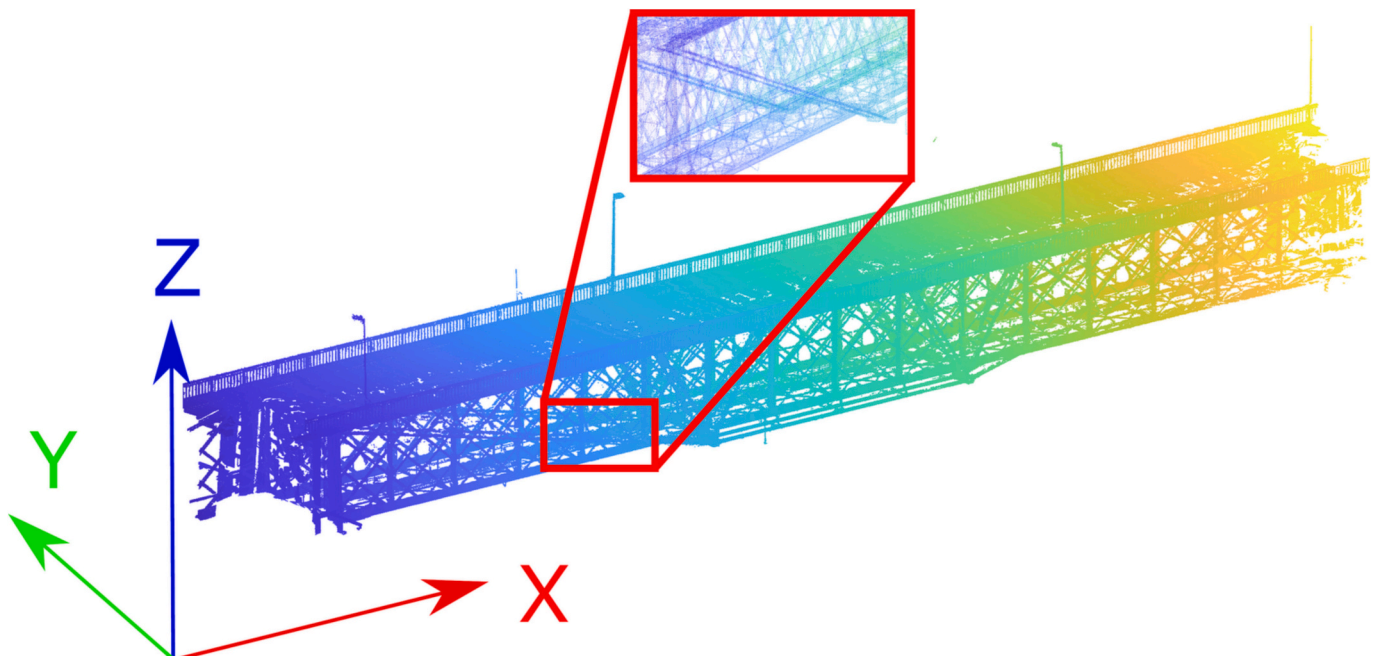


Fig. 12. Voxelised and oriented point cloud.

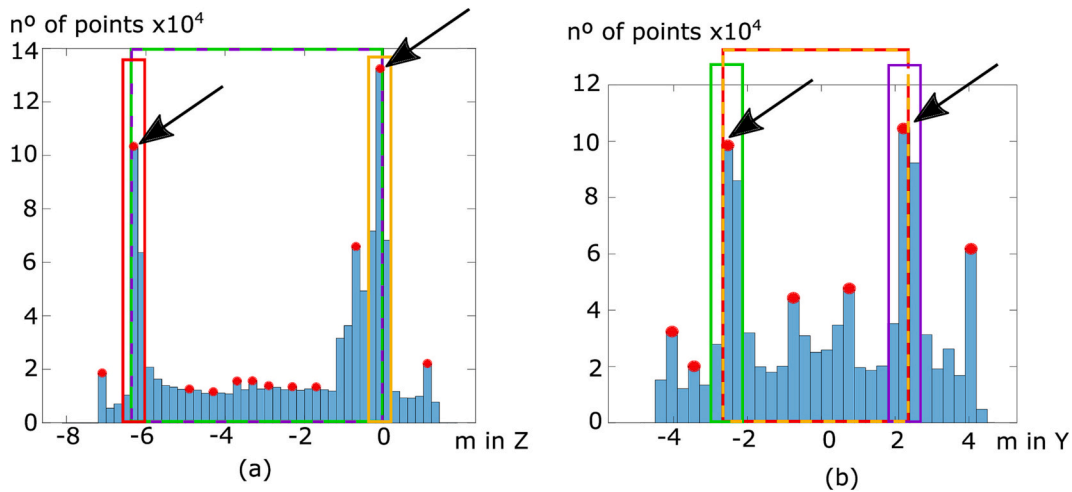


Fig. 13. Histograms for faces segmentation. Red and yellow boxes are horizontal faces. Magenta and green boxes are vertical faces. Arrows show two most occupied coordinates. Red points are local maxima. (a) Z axis coordinate, and (b) Y axis coordinate. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

[28] used voxelisation to detect buildings. Other studies have used voxelisation to extract information from road and railway point clouds [29,30]. Hereafter, whenever a point cloud is mentioned, a reference is made to the voxelised and denoised raw point cloud. Whenever points in this point cloud are mentioned, the centre of mass of their voxels is referred to.

Two additional steps were applied to facilitate the segmentation process. First, the point cloud is denoised by removing isolated points that do not provide relevant information about any element. An isolated point is defined as a point whose mean distance to its  $k$  nearest neighbours is greater than one standard deviation from the mean of the average distance between each point to its  $k$  nearest neighbours. Second, the point cloud was oriented by representing the coordinates in the directions of the eigenvectors obtained from the principal component analysis (PCA) applied to the point cloud. If the alignment of the road of the bridge is available, it is used to calculate the eigenvectors rather than the point cloud, because it contains more reliable information about the orientation of the bridge. Hereafter, the 1st, 2nd and 3rd eigenvectors computed in this step are referred to as the X, Y, and Z axes, respectively.

PCA is an orthogonal linear transformation that transforms data into

a new coordinate system. This new coordinate system was defined by the eigenvectors of the covariance matrix of the point cloud. By applying PCA to spatial data, eigenvectors provide information on the direction of data dispersion. Moreover, the eigenvalues of the covariance matrix provide information regarding the degree of dispersion in each principal direction.

In this study, when PCA was applied, the eigenvalues were recalculated according to Eq. (1) and sorted from highest to lowest. The eigenvectors are sorted according to their eigenvalues.

$$\lambda_i = \frac{\lambda_{0i}}{\sum \lambda} \tag{1}$$

$\lambda_{0i}$ : original eigenvalue  $i$ .

$\lambda_i$ : recalculated eigenvalue  $i$ .

Fig. 12 shows the pre-processed point cloud for case study I.

### 3.3. Faces sectioning

The first segmentation process involved sectioning of the bridge faces. To do so, the local maxima of the Z and Y coordinate histograms,

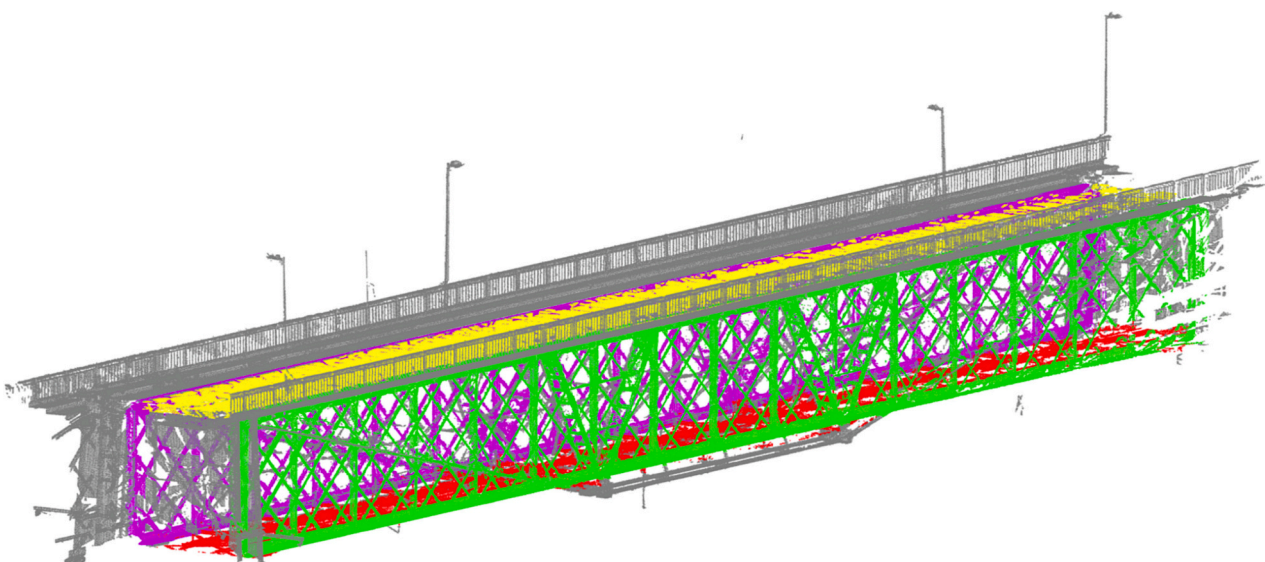


Fig. 14. Voxelised point cloud segmented by faces using the histogram and the colours of Fig. 13.



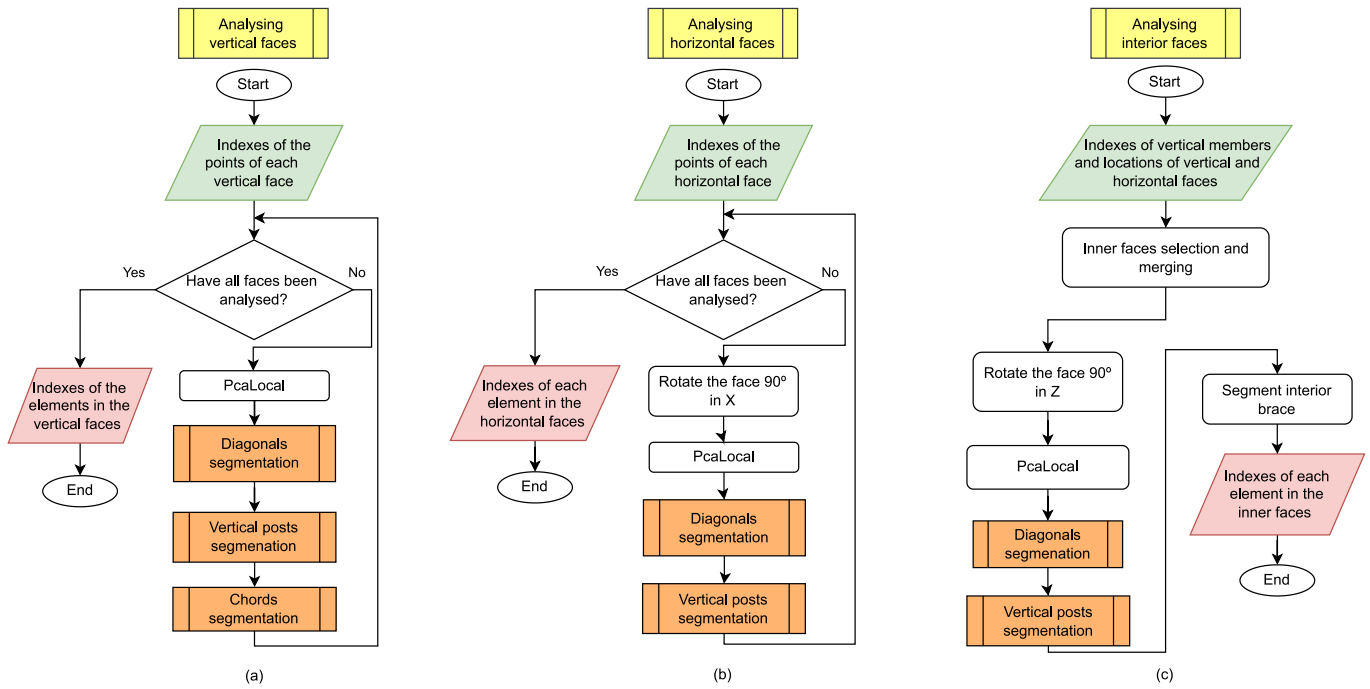


Fig. 15. Faces analysis workflows. (a)Analysing vertical faces, (b)Analysing horizontal faces, and (c) Analysing interior faces.

shown in Fig. 13(a) and Fig. 13(b), were calculated with bin widths *binWidthZ* and *binWidthY*, respectively. The maximum values correspond to the locations of the face. In addition, the limits of a face in a plane are determined by the locations of perpendicular faces. For example, a vertical face limits the locations of horizontal faces up and down. If a bridge does not have a face, for example, the top horizontal face, the histogram also recognises a maximum at the upper end of the vertical face because of the concentration of chord points. For bridges with a deck in the middle of the girder, the first maximum in Z corresponds to the deck location.

To select the points of each face, those points were close to their calculated locations, considering a face width equal to *widthVertFace* and *widthHorFace* for the vertical and horizontal faces, respectively. Additionally, these points must lie between the maximum locations on the histogram along the other axis. For instance, as shown in the red boxes in Fig. 13(a) and Fig. 13(b), the points of the bottom horizontal face are those close to the lower maximum location in Z and between the locations of the vertical faces, which are the maximum locations in Y. Fig. 14 illustrates the bridge segmented into its faces, where the bottom horizontal face used as an example is in red.

### 3.4. Analysing faces

Face-sectioning simplifies the segmentation of the bridge, allowing each face to be analysed independently. The workflow applied to each type of face is shown in Fig. 15. These workflows were designed to be as similar as possible, reorienting the faces to use the same metrics, and applying specific functions to search for the elements present in them. The functions used for the segmentation of each item type, represented by orange rectangles in the diagrams, are described in section 3.5.

#### 3.4.1. Analysing vertical faces

The vertical faces are green and magenta, as shown in Fig. 14. These faces had two types of items: diagonals and vertical posts. Chords were also included in the analysis. Fig. 15(a) shows the workflow used to analyse vertical faces.

This segmentation process analyses all the faces sequentially and independently. First, points on the face under study were selected. The local PCA was then computed. Local PCA consists of applying PCA to the neighbourhood of each point. This neighbourhood includes all points closer than *distLPCA* to the point under analysis and the point itself. This analysis allows us to select points whose neighbourhood dispersion is required using Eq. (2).

$$\Lambda > \Lambda_{\min} \& \Lambda < \Lambda_{\max} \& V > V_{\min} \& V < V_{\max} \tag{2}$$

Being:

$\Lambda$ : eigenvalue array of applying local PCA to a point.

$\Lambda_{\min}$ : minimum eigenvalues defined as threshold.

$\Lambda_{\max}$ : maximum eigenvalues defined as threshold.

$V$ : eigenvector matrix for applying local PCA to a point.

$V_{\min}$ : minimum eigenvector defined as the threshold.

$V_{\max}$ : maximum eigenvector defined as the threshold.

$\otimes <$ : less than elementwise.

$\otimes >$ : greater than elementwise.

The results of this analysis were used as inputs for algorithms for segmenting diagonals, vertical posts, and chords as explained in sections 3.5.1, 3.5.2, and 3.5.3, respectively.

Fig. 16 shows a vertical face segmented using this methodology.

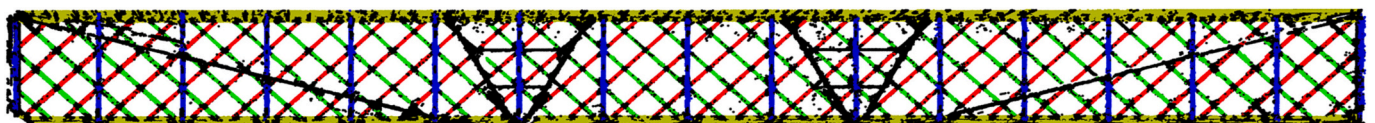


Fig. 16. Vertical face segmented. Diagonals in red and green, vertical posts in blue, chords in ochre and no segmented points in black. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

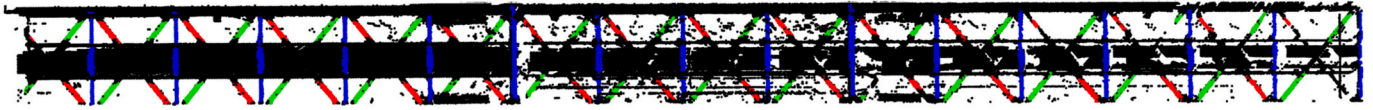


Fig. 17. Horizontal face segmented. Bottom lateral braces in red and green, struts in blue, and points not segmented in black. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

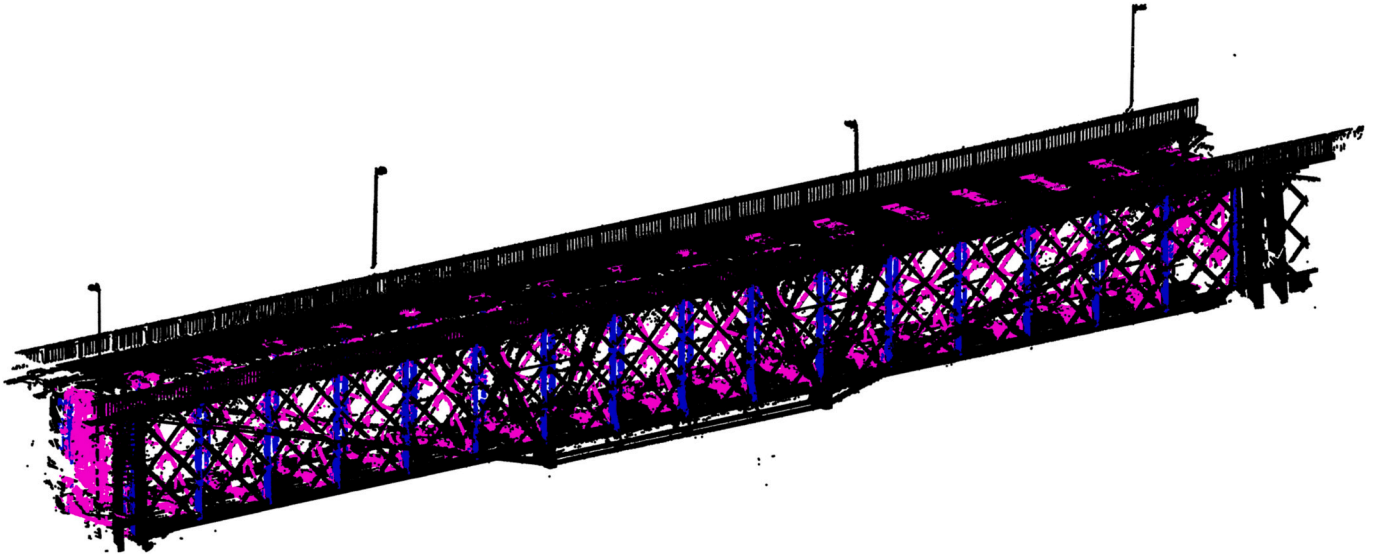


Fig. 18. Voxelised point cloud with interior faces segmented. Interior faces in magenta and vertical members in blue. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

### 3.4.2. Analysing horizontal faces

The horizontal faces are the red and yellow faces shown in Fig. 14. The yellow face is the asphalt of the bridge; therefore, it was not analysed. The red face was analysed following the workflow shown in Fig. 15(b). This face has two types of items: bottom-lateral braces and struts. In addition, there is a gangway close to this face which, in the face-sectioning process, is sectioned as part of the bottom horizontal face owing to its proximity to it. However, the segmentation of this gangway is not an objective of this work, and thus, its points are not analysed.

This process is like that described above in section 3.4.1. The first process consisted of rotation of the face under study by  $90^\circ$  along the x-axis (as defined in Fig. 12). Therefore, the horizontal face is reoriented vertically. This allows us to continue the process as a vertical face, segmenting the bottom lateral braces using the algorithm explained in section 3.5.1, and the struts with the algorithm explained in section 3.5.2, because a rotated strut can be segmented in the same manner as a vertical post.

Fig. 17 shows a vertical face segmented using this methodology. This figure shows not only points segmented as struts or bottom lateral braces, but also non-segmented points such as points that belong to the chords (already segmented in the vertical face analysis), gangways, or errors in the process.

### 3.4.3. Analysing interior faces

The interior faces were the last faces analysed. This analysis followed the workflow shown in Fig. 15(c). Unlike the vertical and horizontal faces, the interior faces were not sectioned in the same manner. Thus, the segmentation of the bridge interior faces uses vertical member positions. Each interior face is formed by the points between two vertical members with the same X position, considering that the width of the face is equal to *widthInnerFace*. Fig. 18 shows the interior faces of the bridge.

Once the interior faces were sectioned, their elements were segmented into interior lateral braces, interior braces, and vertical posts.

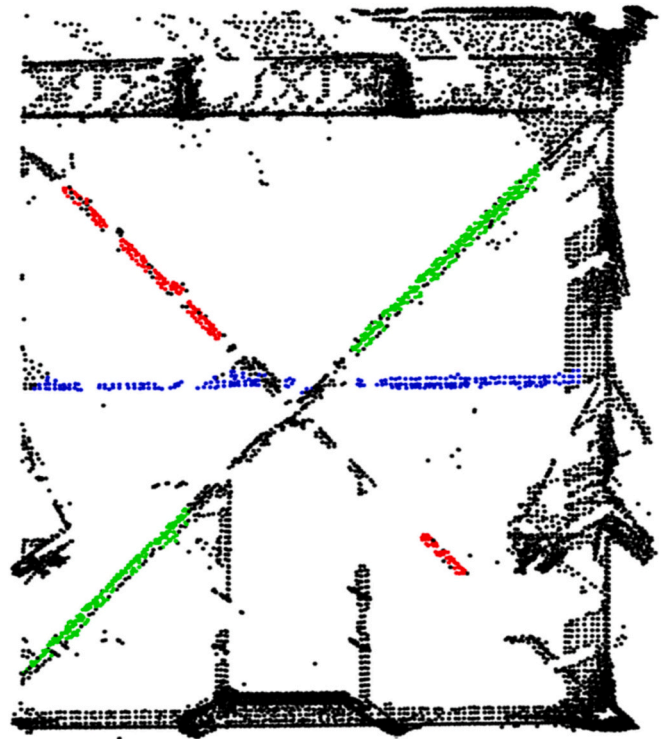


Fig. 19. Interior face segmented. Interior lateral braces in red and green, interior braces in blue and points not segmented in black. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

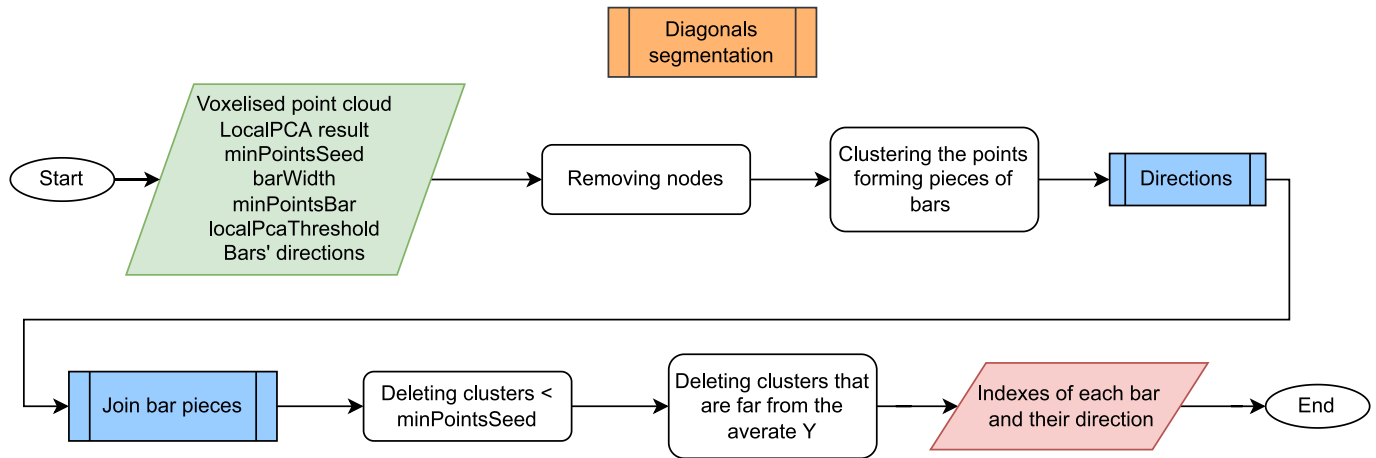


Fig. 20. Segmentation workflow for diagonals.

The workflow is like that of the other faces, but with the following differences:

First, all faces were segmented jointly instead of individually, as for horizontal and vertical faces. Second, a local PCA was applied, as for the other faces. Third, as in the horizontal faces, the interior faces were rotated by  $90^\circ$ , but this time on the Z-axis, to reorient them as vertical faces.

The interior lateral braces were then segmented using the method described in Section 3.5.1, with the difference that the last step of this process, Y-axis filtering, was not applied. There are two reasons. First, this was unnecessary because there were no elements external to the truss near these faces. Second, these faces are shorter than the others; therefore, defining the plane of each interior brace to filter out the elements that move away from it is complicated.

Vertical posts were then segmented by applying the method described in section 3.5.2, as in the segmentation of other types of faces.

Finally, the interior braces were not segmented using the method described in section 3.5.3 because there was only one member per face. The procedure was as follows. First, using the local PCA results, points with horizontal dispersion that are not in the extremes of Z are chosen. A histogram in Z is then calculated, and points close to the maximum bin are selected. The resulting points are segmented into interior braces. The segmented interior face is shown in Fig. 19.

### 3.5. Truss element segmentation processes

This section describes the methodologies used in section 3.4 to segment each type of element, using the methods represented by orange rectangles in Fig. 15. These methods are designed to segment the elements present on a vertical face. However, they are also used to segment

analogous elements into horizontal and interior faces after reorienting them as vertical faces.

#### 3.5.1. Segmentation of diagonals

This section describes the method used for segmenting not only diagonals but also bottom and interior lateral braces if their faces are oriented as vertical faces. Fig. 20 shows the workflow of this process.

The diagonals are elements joined by nodes. Because the objective is to recognise the points belonging to each bar, the first step involves removing these nodes. In this step, vertical posts and chord members are also removed. To do so, the local PCA result, which is an input, is used to select those points that have a linear dispersion, and that this dispersion is neither completely in X nor Z. So, Eq. (2) is applied being  $\lambda_{min} = \text{minEigvalDiag}$  and  $v_{max} = \text{maxEigvectDiag}$ . The points that matched the equations were selected. The results are shown in Fig. 21. This method removes vertical posts and chords because they are dispersed along the X- or Z-axes. Nodes were removed because their dispersion was planar rather than linear.

These points were then clustered using DBSCAN [21] with epsilon and minimum point parameters equal to  $\text{epsilonDiag}$  and 1, respectively. The results are shown in Fig. 22, in which groups of bars are formed.

Once the points were clustered into bars, they were grouped to form a unique cluster for each diagonal bar. To achieve this, it is necessary to know the directions of the bars. There are two directions because the truss comprises diagonals that point up and diagonals that point down. The calculation of the directions follows the workflow shown in Fig. 23. First, the direction of each bar was calculated by applying a PCA to its points. The 1st eigenvector is the direction of the bar. The directions are then used as the input of DBSCAN [21] for clustering, with epsilon and

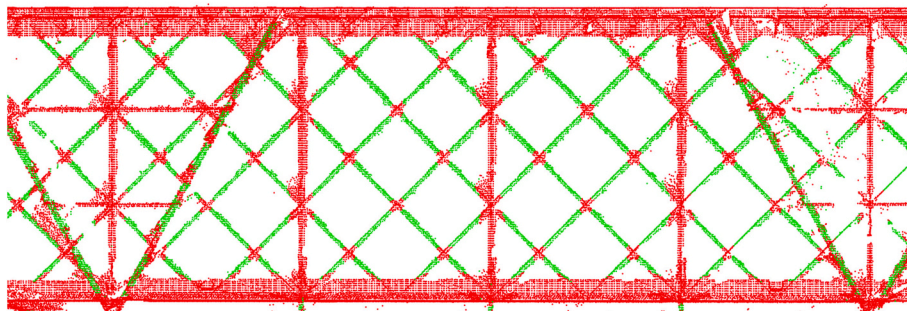


Fig. 21. Result of the method for removing points of nodes, chords, and vertical members. Removals are the red points. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



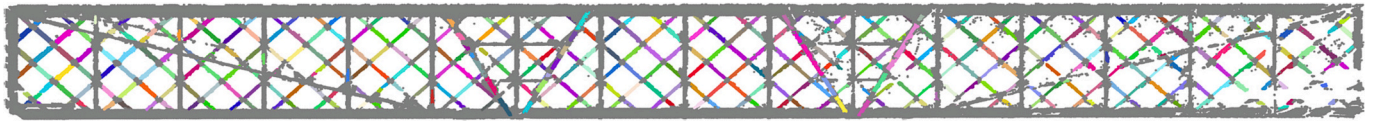


Fig. 22. Vertical face segmented in pieces of possible diagonal. Each cluster is in a different colour. Gray points belong to nodes, vertical posts, and chords.

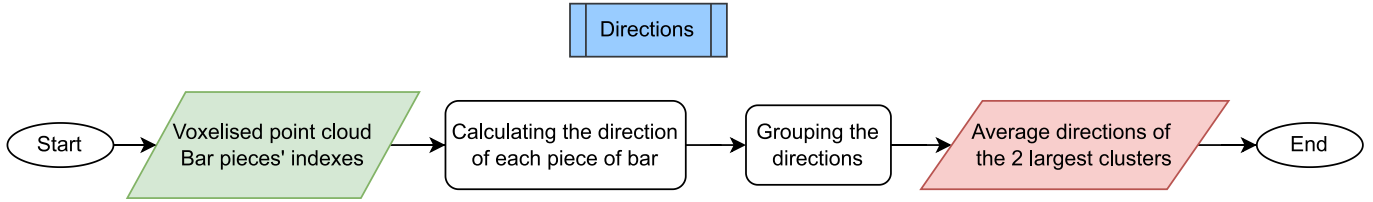


Fig. 23. Calculation of the direction workflow of lateral diagonals.

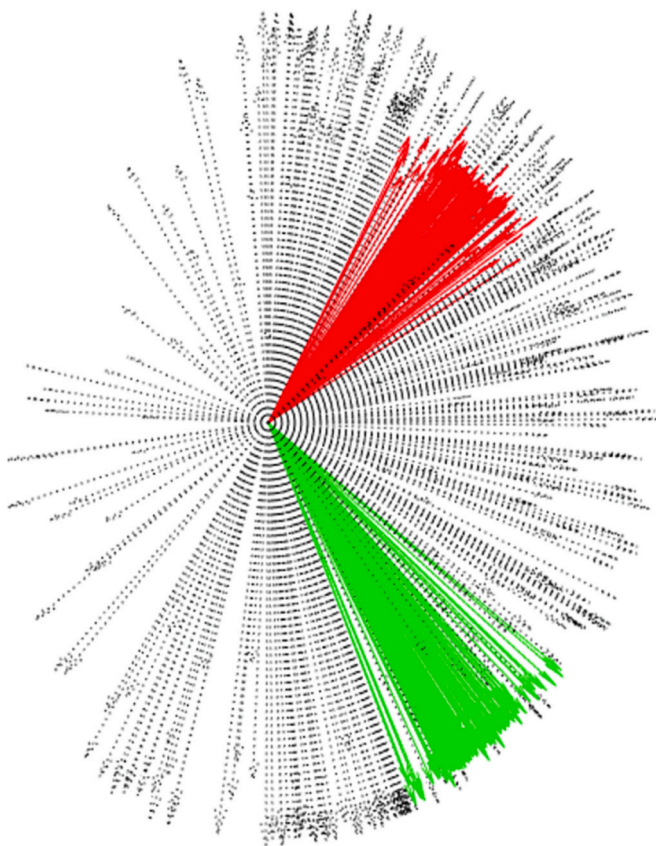


Fig. 24. Directions of each of the possible pieces of diagonals. Coloured in red and in green the directions which form the two major clusters. The mean of the red directions is the direction of the bars that point up. The mean of the green directions is the direction of the bars that point down. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

minimum point parameters equal to *epsilonDirections* and *min-PointsDirections*. This process is illustrated in Fig. 24. The mean directions of the groups with more points correspond to the two directions of the lateral members.

Once the directions of the diagonals are known, pieces of the bars are joined according to the workflow shown in Fig. 26. This process has two major stages: the removal of clusters with fewer points than *minSeedDirections*.

In the first stage, the clusters are assigned to one of the two diagonal directions or none. This was performed by calculating the direction of each cluster as the first eigenvector of the PCA result and comparing it with the diagonal directions. They were assigned a direction if the dot product between them was higher than *toleranceDirections*. The results are shown in Fig. 25.

In the second stage, the clusters were merged. For this purpose, the clusters assigned in the same direction were studied. These clusters were sorted according to the number of points. Starting from the largest bar, the bounding box of the bar under study was defined. This box was centred at the centre of mass of the cluster, without a limit in the direction assigned to that cluster (calculated with the algorithm of Fig. 23), and with a width equal to *barWidth*. Next, clusters in the same direction as any point inside the box were merged. An example of this process is shown in Fig. 27. Clusters that were not assigned any direction were not considered.

Finally, the clusters were filtered. Clusters further than *laterallyYDiag* from the Y-centre of these clusters were deleted. The final clusters are diagonals, bottom lateral braces, or interior braces depending on the face to which they belong.

### 3.5.2. Segmentation of vertical posts

This section describes the method used for segmenting not only vertical posts, but also struts and interior braces if their faces are re-oriented as vertical faces. These items are straight vertical bars equidistant from each other. Fig. 28 shows the workflow of this process. It is important to note that this process was applied after the method described in section 3.5.1, and points that were already segmented were not considered in this analysis.

The first step consists of selecting points whose neighbourhoods have

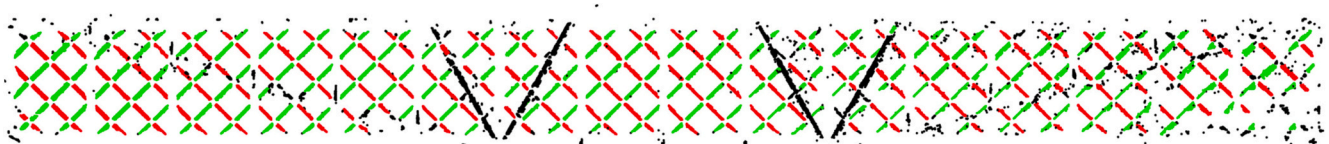


Fig. 25. Selection of pieces of bars by its direction. Green and red points are members of clusters assigned to a direction. Black points are members of clusters with a different direction. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Join bar pieces

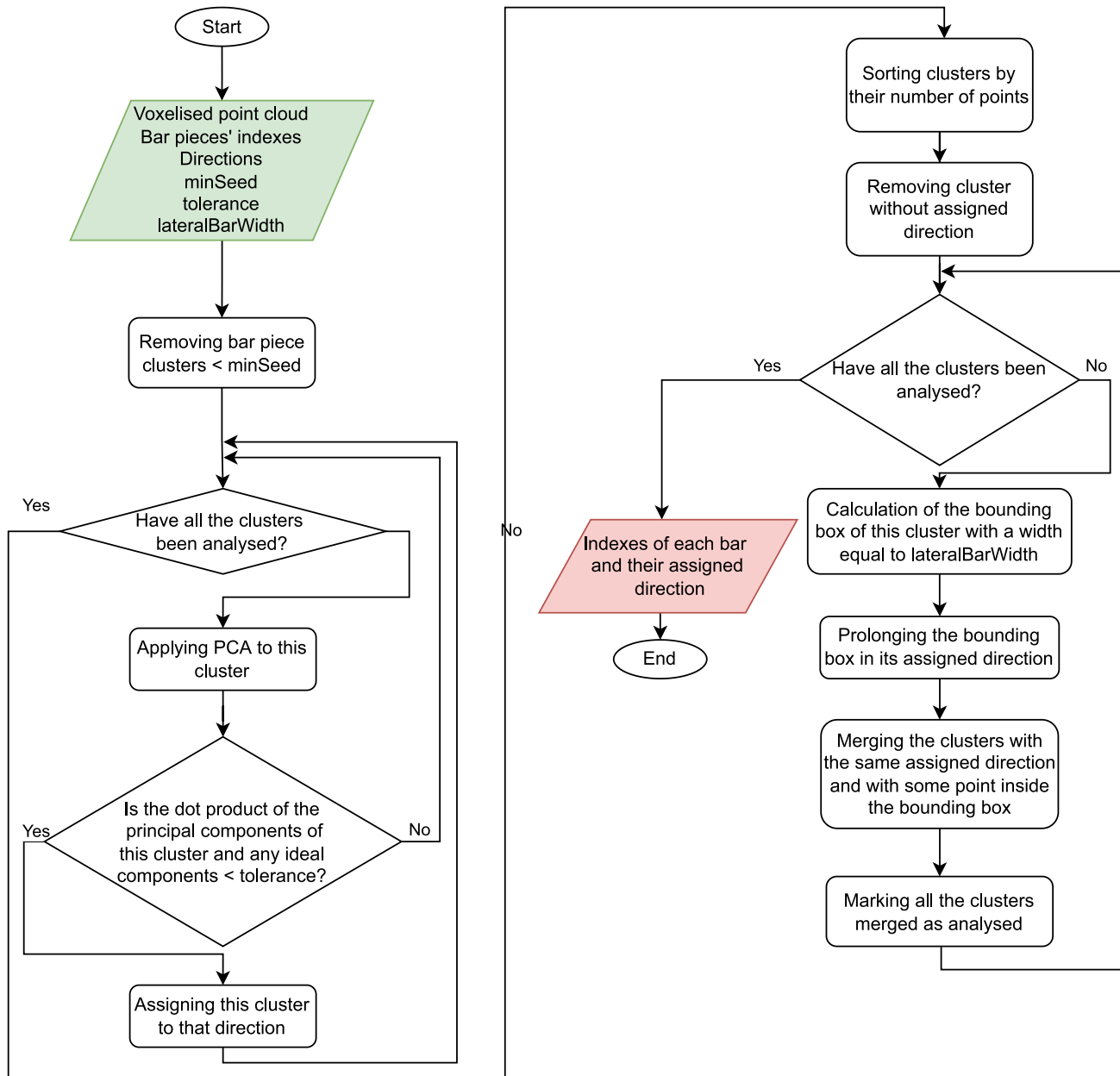


Fig. 26. Merging lateral diagonal pieces workflow.

vertical dispersion. Using the local PCA result, points that match Eq. (2) with  $v_{min} = \mathbf{minEigVectVert}$  are selected.

Once the vertical points are selected, a histogram of their X position with a bin width of  $\mathbf{binWidth}$  is calculated. The objective is to find the location of each bar, knowing that there are maxima in those positions and that those locations are equidistant. To calculate these locations, the methodology shown in Fig. 29 is applied and named *EquidistantHistPeaks*. This process consists of finding histogram maxima that are equidistant. For this, all local histogram maxima are considered. The variance in the distances between consecutive maximum locations is then calculated. If the variance is greater than  $\mathbf{maxVarPeaks}$ , the smallest maximum is removed, and the variance is recalculated. When the variance is sufficiently small, the locations of the maxima are the

locations of the vertical posts. Fig. 30 illustrates the histogram processing.

Finally, the points in the computed locations, considering a bar width equal to  $\mathbf{vertBarWidth}$ , are clustered, forming vertical posts, struts, or interior braces, depending on the face to which they belong. An example of the segmentation of these elements is shown in Fig. 31.

### 3.5.3. Segmentation of chords

Fig. 32 shows the workflow of this process. This section describes the method used to segment chords. Chords are horizontal members parallel to the X-axis of the reoriented bridge that delimit the truss. For this reason, the first steps are like those described above in section 3.5.2: selecting points that match Eq. (2) with  $v_{min} = \mathbf{minEigVectChord}$ , and

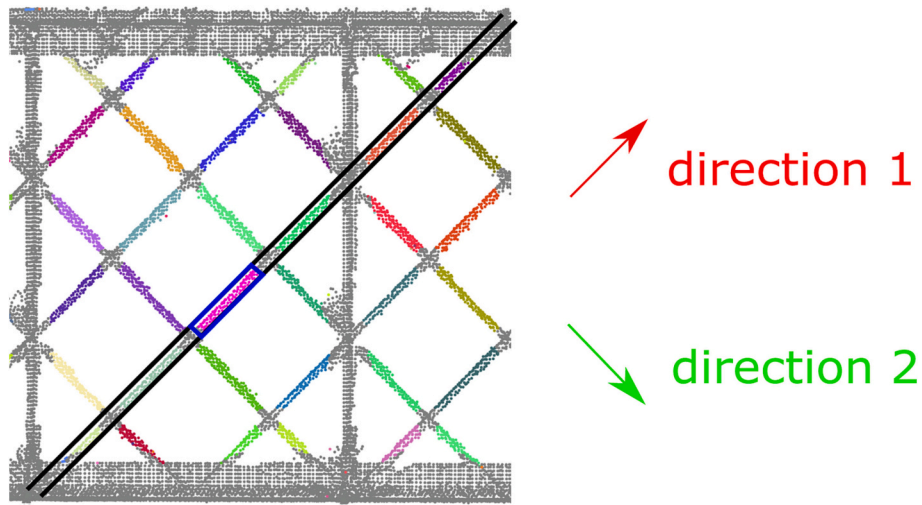


Fig. 27. Merging pieces of diagonals. Each piece is coloured differently. The studied cluster is pink, inside the blue box. The blue box is extended in direction 1, painted in black. All the clusters with any point in the black box and assigned to direction 1 are merged, forming a diagonal. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

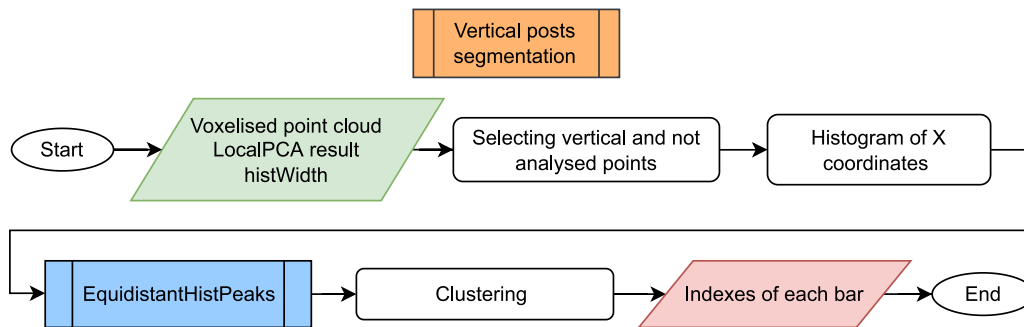


Fig. 28. Segmentation workflow of vertical members.

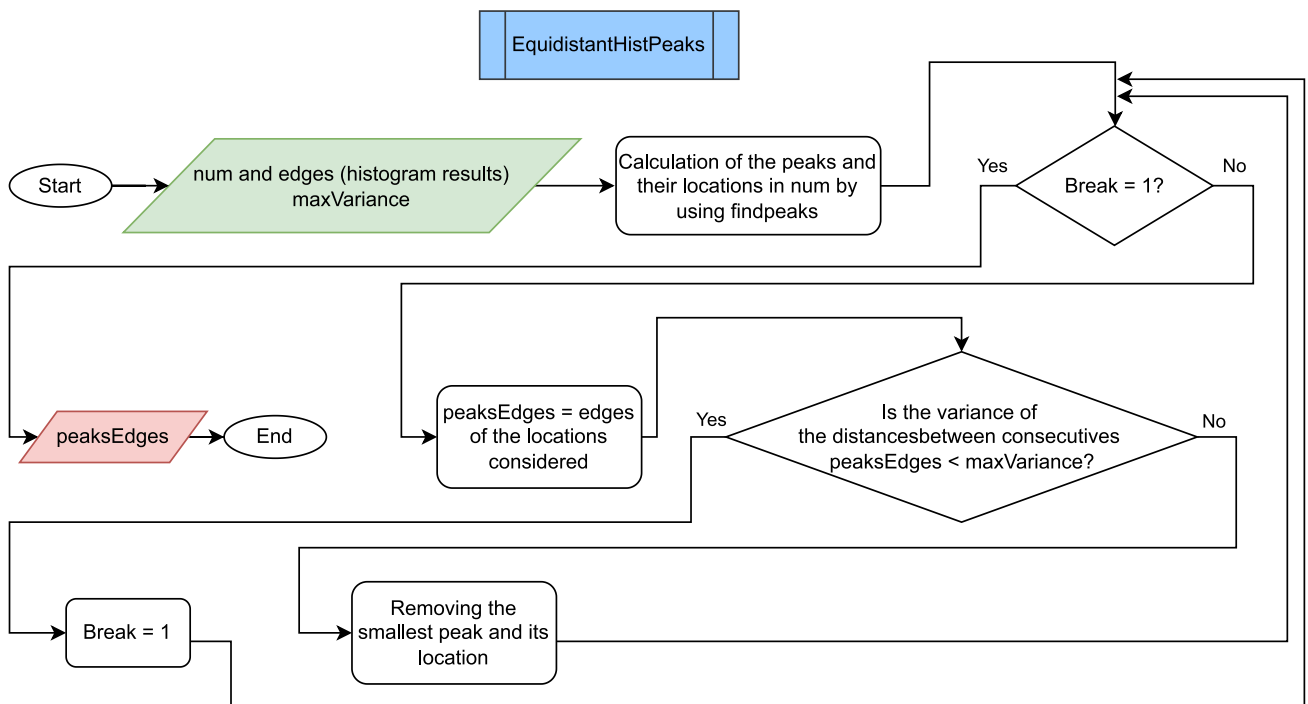


Fig. 29. Equidistant histogram peaks workflow.



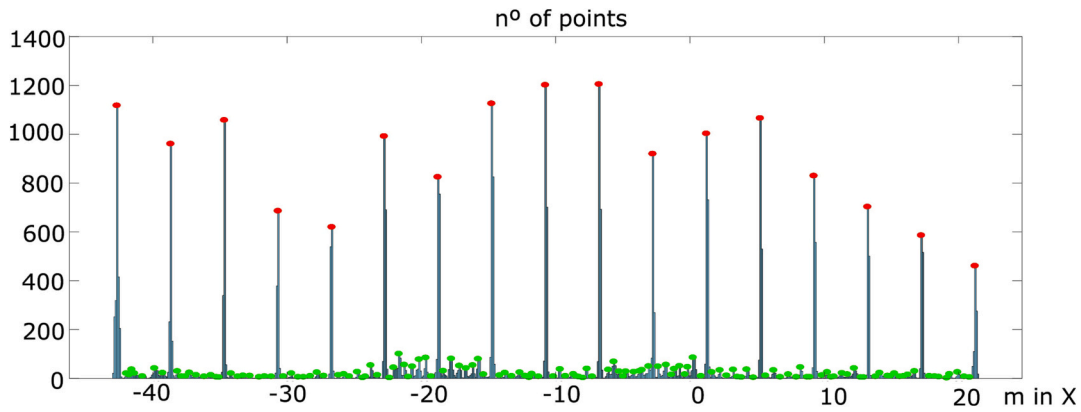


Fig. 30. Histogram for the location of vertical members. Peaks of the first iteration in green and peaks of the last iteration in red using Fig. 29 workflow. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

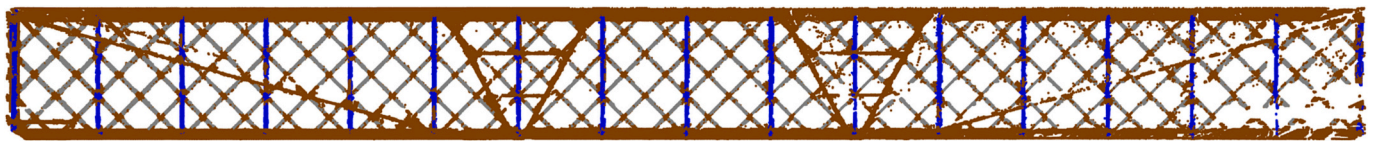


Fig. 31. Vertical posts segmented. Vertical posts in blue. Points analysed and not segmented as vertical posts in brown. Points that are not analysed because they are segmented in a previous step as a different element in gray. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

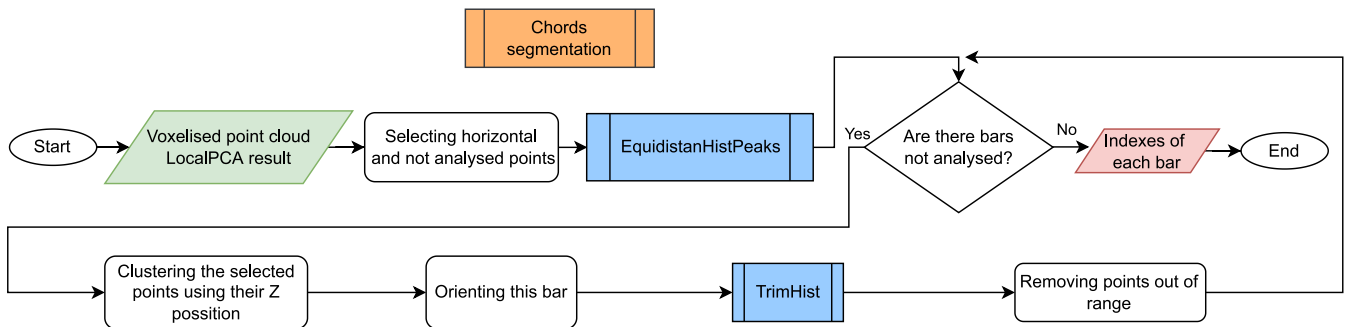


Fig. 32. Segmentation workflow of chords.

calculating the Z position of each chord using *EquidistantHistPeaks*. It is important to note that this process is applied after the methods described in sections 3.5.1 and 3.5.2, and the points already segmented are not considered in this analysis.

For each bar, the points closer to the location of the chord member are clustered, considering a chord width in Z equal to *chordWidth*. The cluster is then oriented using a modified version of PCA [30] to be denoised in Y. The objective is to remove points that are not part of the bar. To do so, a histogram of the Y position with a bin width equal to *binWidth* is generated. This is used to trim the points outside the bar limits using the workflow described in Fig. 33, and named *TrimHist*. This method involves sequentially incrementing the number of bins considered, starting from the most populated bin of the histogram. For each increment, the ratio of the points considered to the total is computed. This increment is repeated until the difference between the actual ratio and the previous ratio is lower than *histThreshold*. An example of a trim histogram is shown in Fig. 34(a) and a denoised chord is shown in Fig. 34(b).

Note that in Fig. 34(a), the points inside the red box are those selected using the method described in Fig. 33.

#### 4. Results

The values of the parameters presented in section 3, which are listed in Table 1, are based on the experience of the authors and have been tuned on the basis of experiments. The same values are used in all case studies.

The *g* parameter value requires a compromise between down sampling and accuracy. It is an empirical value that is determined using the experience of the authors and the magnitude of the bridge elements. *binWidth* must be greater than *g* because it is used to find groups of voxels in the point cloud. *binWidthY* and *binWidthZ* are used to determine the location of the faces in a histogram; therefore, they must be smaller than the face width. *distLPCA* must have a value that allows for the differentiation of the truss parts; for instance, to differentiate nodes from bars. The eigenvalue parameters are adjusted depending on whether the cluster is linear, planar, or volumetric. The eigenvector parameters define only the 1st eigenvector. These indicate the main directions of the segmented group.

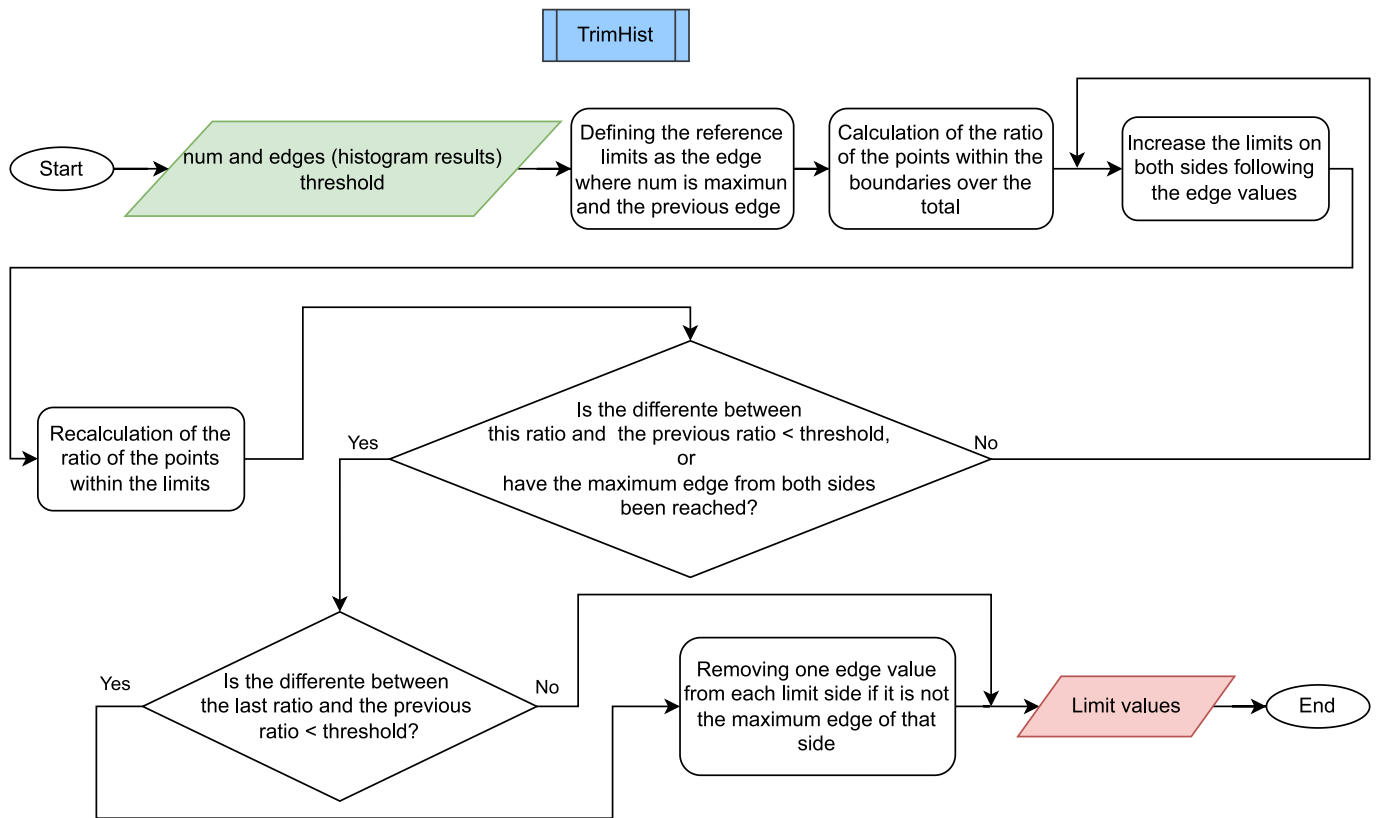


Fig. 33. Trim histogram workflow.

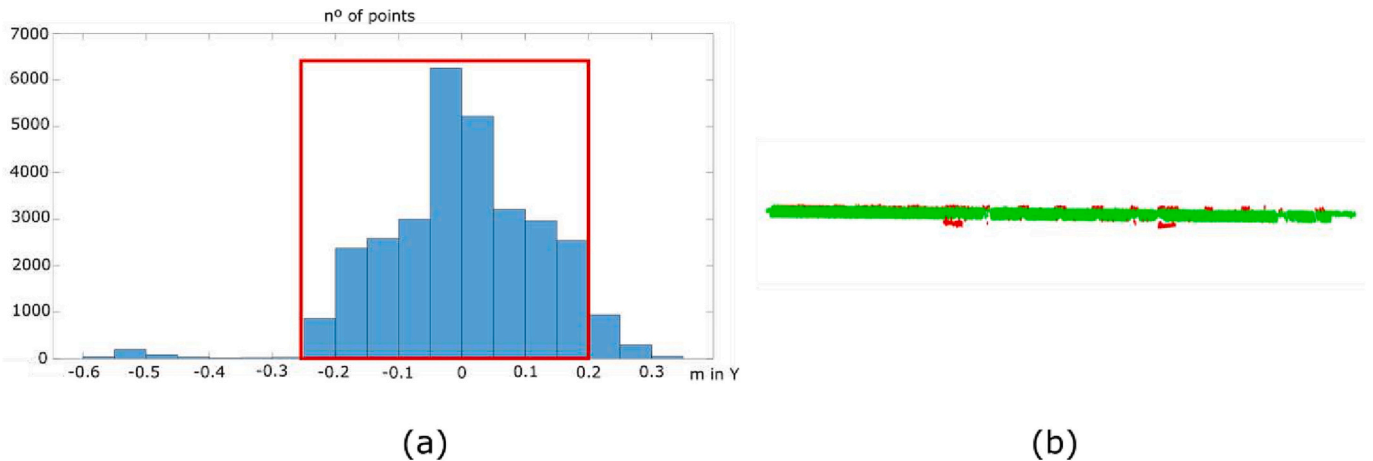


Fig. 34. (a) Histogram of Y locations in chord member segmentation, (b) the segmented chord where the points selected are in green, and the discarded in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.1. Case study I

The specific parameters used in the case study are listed in Table 2. These parameters are based on the dimensions of the bridge components. These are the widths of its faces and the widths of the cross-sections of its members, which were manually measured in the point cloud.

The elements detected are shown in Fig. 35. The analysed truss bridge had a total of 274 elements comprising 17 struts, 32 bottom lateral braces, 17 interior braces, 34 interior lateral braces, 34 vertical posts, 136 diagonals, and 4 chords. The following errors are detected:

Fig. 35 shows that the leftmost strut is not detected and is a false

negative.

On the bottom horizontal face, an external element is segmented as a bottom lateral brace. Fig. 36(a) shows the error.

In the interior face of Fig. 36(b), there are three false positives. This face is close to a pillar, and a part of it, which is segmented, has interior lateral braces.

On the interior face shown in Fig. 36(c), which is the interior face close to the other pillar, there is one false negative because an interior lateral brace is not detected.

On another interior face, as shown in Fig. 36(d), there is an interior lateral brace shown, which is a false positive.

Moreover, the detected elements are partially segmented. However, the intention is not to achieve complete segmentation, but to detect each

**Table 1**  
Values of the parameters used.

Process	Parameter	Value	Unit	
Common parameters Pre-processing	<i>binWidth</i>	2 <i>g</i>	m	
	<i>g</i>	0.05	m	
	<i>k</i>	4		
Faces sectioning	<i>binWidthY</i>	<i>widthVertFace</i> /3	m	
	<i>binWidthZ</i>	<i>widthHorFace</i> /3	m	
Analysing faces	<i>distLPCA</i>	6 <i>g</i>	m	
Segmentation of diagonals	<i>minEigvalDiag</i>	(0.7 - -)		
	<i>maxEigvectdiag</i>	$\begin{pmatrix} 0.9 & - & - \\ - & - & - \\ 0.9 & - & - \end{pmatrix}$		
	<i>epsilonDiag</i>	4/3 <i>g</i>		
	<i>epsilonDirections</i>	2 <i>g</i>		
	<i>minPointsDirections</i>	10		
	<i>minSeedDirections</i>	1.5/ <i>g</i>		
	<i>toleranceDirections</i>	0.97		
	<i>lateralYDiag</i>	4/3 <i>barWidth</i>	m	
	Segmentation of vertical posts	<i>minEigvectVert</i>	$\begin{pmatrix} - & - & - \\ - & - & - \\ 0.9 & - & - \end{pmatrix}$	
		<i>maxVarPeaks</i>	0.1 <i>g</i>	
Segmentation of chords	<i>minEigvectChord</i>	$\begin{pmatrix} 0.9 & - & - \\ - & - & - \\ - & - & - \end{pmatrix}$		
	<i>histThreshold</i>	0.05		

**Table 2**  
Parameters used in segmentation of case study I.

Parameter	Value [m]	Parameter	Value [m]
<i>barWidth</i>	0.1	<i>widthVertFace</i>	0.9
<i>vertBarWidth</i>	0.3	<i>widthHorFace</i>	0.6
<i>chordWidth</i>	1	<i>widthInnerFace</i>	0.9

item at least partially, whose main direction matches that of the item to which it belongs. As explained in the methodology section, instances are selected based on their direction. This implies that each detected instance has the same principal direction as the member that it represents. However, it would be interesting to analyse the parts of the truss elements that are not segmented.

Fig. 37(a) shows how the diagonals are completely segmented, except for the nodes, and how the gangway hinders the segmentation of the interior lateral braces in Fig. 37(b). These areas are not segmented because the nodes are removed in the segmentation process to isolate each member of the truss.

On the interior faces, as shown in Fig. 37(c), the same occurs with the nodes on the diagonals; however, some points of the bars are not well recognised. This results from the fact that there are discontinuities in the element itself due to occlusions in the point cloud.

Lastly, it is important to note that the point cloud has occlusions, as is

shown in Fig. 8 and in the images of Fig. 36.

In summary, the segmentation of case study I yielded 272 true positives, 2 false negatives, and 6 false positives.

#### 4.2. Case study II

The specific parameters used to process this case study are shown in Table 3.

The elements detected are shown in Fig. 38.

The truss bridge analysed has 196 elements comprising 12 struts, 14 bottom lateral braces, 24 interior vertical braces, 72 interior lateral braces, 24 vertical posts, 44 diagonals, and 6 chords. The following errors are detected:

A lateral brace is not detected, as is shown in Fig. 39.

In the horizontal face, 5 bottom lateral braces are not detected. Ten struts are not detected on the same surface. Fig. 40 shows the false negatives.

In the interior faces, 25 interior lateral braces and one vertical post are not detected. Fig. 42 shows the false negatives.

Moreover, as in case study I, the detected elements are not completely segmented; however, the objective of the segmentation is achieved because the principal direction of these elements are correct.

Lastly, it is important to note that the point cloud has occlusions, as is shown in Figs. 10, 39, 40, and 42. This makes it impossible to segment the two chords under the bridge deck.

In summary, the segmentation of case study II yielded 154 true positives and 42 false negatives.

### 5. Discussion

The main objective of this study is to develop an automatic instance segmentation method for truss-bridge point clouds. To the best of our knowledge, the justification for this objective is based on the absence of a segmentation process for truss bridges. Therefore, this study represents the publication of code and data as the first benchmark for the segmentation of truss bridge point-cloud instances.

The specific objective of this algorithm is to at least partially detect every set of points whose main direction matches that of the item to which it belongs. This objective allows the determination of the positions of the bridge nodes and the elements connected to them, as demonstrated in the continuation of this study [31]. This information is essential for modelling purposes. These nodes are critical for determining the geometry and structural behaviour of the bridge, and they help ensure that the bridge can withstand the loads and stresses it will experience over its lifetime. The proper modelling of these nodes is important for accurately predicting the performance and safety of truss bridges.

The results obtained demonstrate the capacity of the algorithm to achieve the proposed objective, which was tested using two case studies.

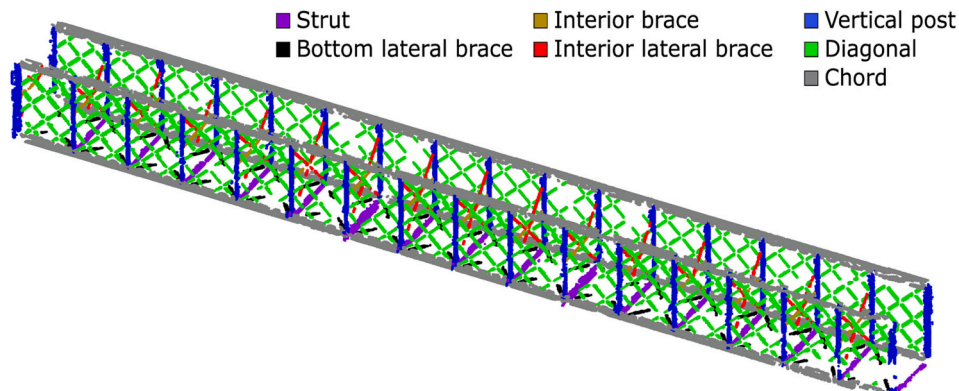


Fig. 35. Case study I bridge segmented.



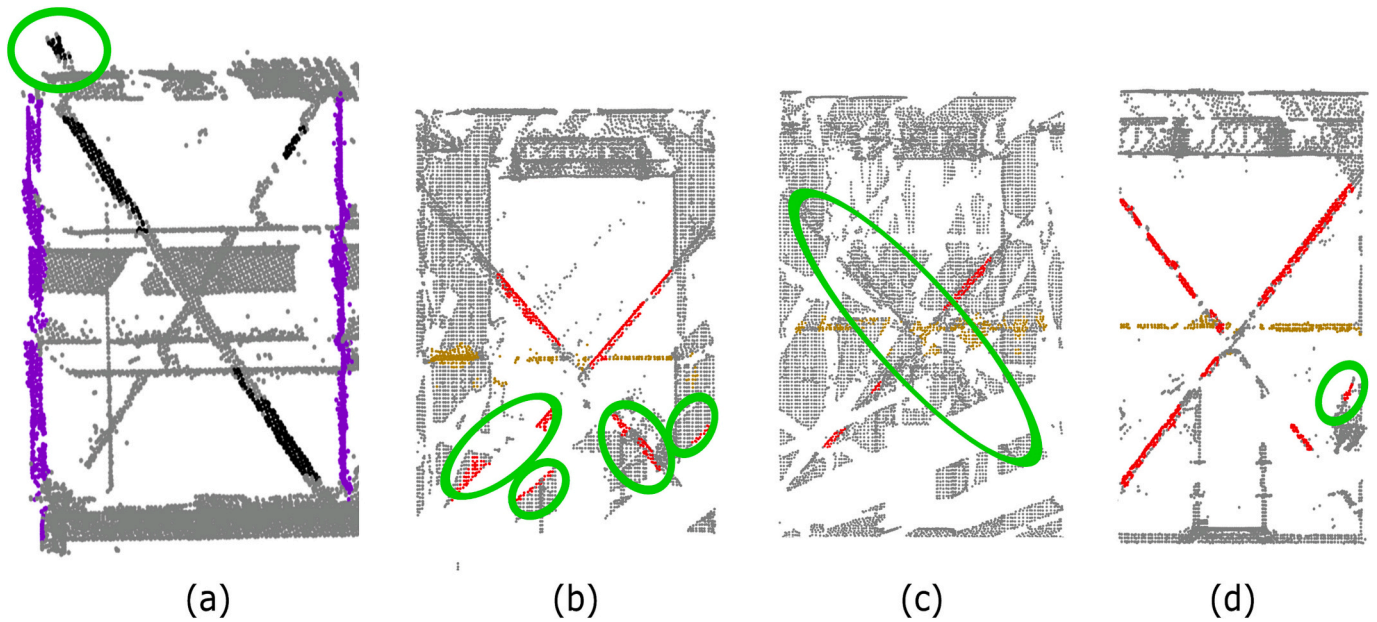


Fig. 36. Errors in case study I bridge segmentation. Errors are shown in a green circle. (a) False negative in the bottom horizontal face, (b) false positives in an interior face close to a pillar, (c) false positives in an interior face close to the other pillar, and (d) false positive in an interior face. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

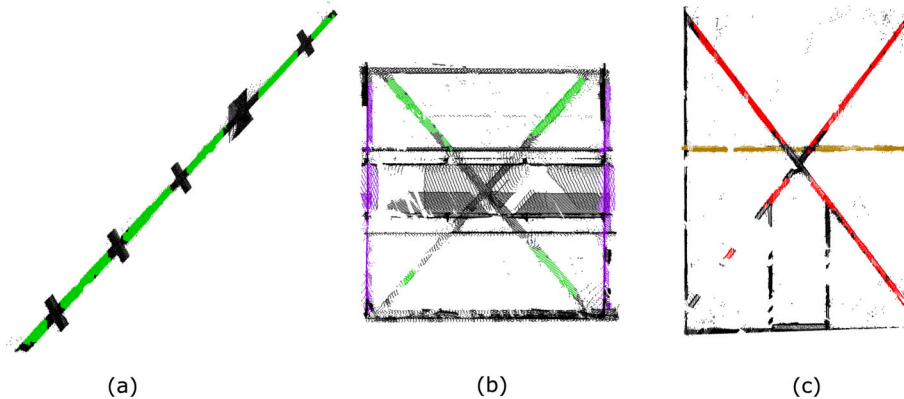


Fig. 37. Segmented elements from different faces with no segmented points in black. (a) Diagonal in green, (b) bottom face with struts in violet and bottom lateral braces in green, and (c) interior face with interior brace in brown and interior lateral braces in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3  
Parameters used in the segmentation of case study II.

Parameter	Value [m]	Parameter	Value [m]
<i>barWidth</i>	0.3	<i>widthVertFace</i>	0.6
<i>vertBarWidth</i>	0.2	<i>widthHorFace</i>	0.6
<i>chordWidth</i>	0.5	<i>widthInnerFace</i>	0.6

In case study I, the good performance of the algorithm can be observed by comparing Fig. 2 and Fig. 35. In this case study, 272 true positives, 2 false negatives, and 6 false positives are obtained. Most errors are caused by occlusions or noise in the point cloud. In fact, all the errors are in the interior faces, where there are more occlusions, and close to the pillars, where the truss elements are sometimes mistaken for pillars. This is shown in Fig. 36.

In case study II, the results are observed by comparing Fig. 5 and Fig. 38. In this case study, 154 true positives and 42 false negatives are identified. Most of the errors are due to the reasons already discussed.

The errors in the interior faces are due to occlusions, as shown in Fig. 42, where the number of occlusions is highly relevant. In addition, the errors in the horizontal face can be attributed to two factors. First, false positives of the struts are caused by an item close to the pillar. *EquidistantHistPeaks* discards strut positions sequentially and discards false struts after discarding several real struts; thus, they are not detected. The results are shown in Fig. 40 and Fig. 41. Second, the error in the bottom lateral brace detection occurs because they do not follow the same direction, as shown in Fig. 40, and the algorithm does not foresee this possibility. However, it could detect 9 of the 14 braces despite occlusions.

It is important to discuss the effects of the point cloud quality and truss topology on the results.

In any data analysis process, the data quality is the first determining factor. For point clouds, particularly in these two case studies, the limiting factor for the information that can be extracted from point clouds is the presence of occlusions. In point clouds, the occlusions are missing points belonging to nonvisible surfaces from any scan position. Occlusions are common in point clouds with such characteristics. Owing

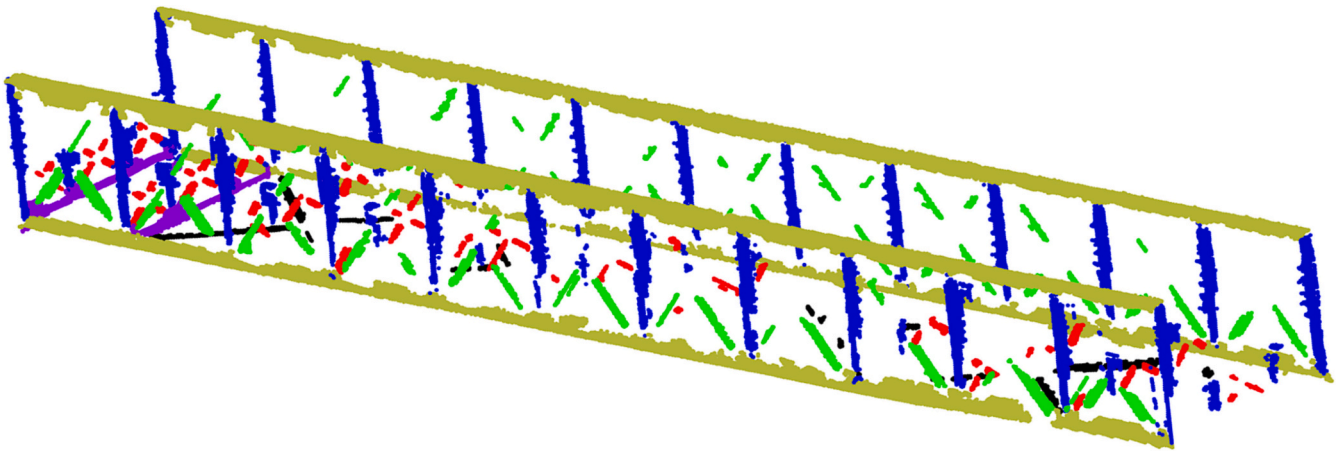


Fig. 38. Case study II bridge segmented.



Fig. 39. Vertical face of case study II bridge segmented. The error is marked with a red circle. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

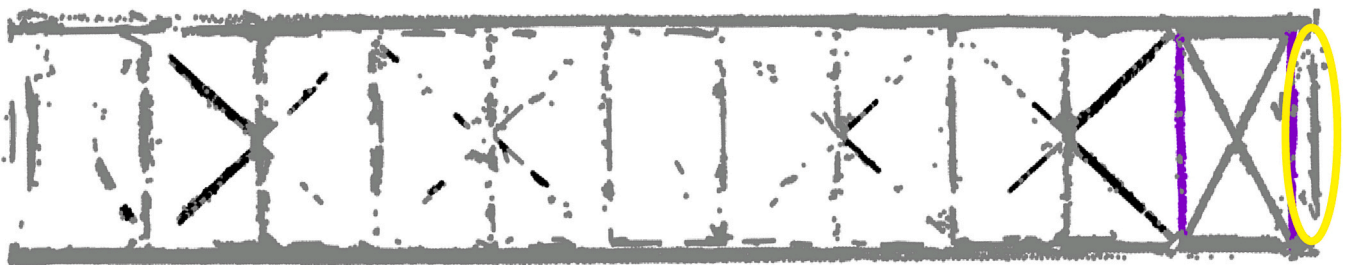


Fig. 40. Horizontal face of case study II bridge segmented. The item inside the yellow circle is the cause of the failure in the strut segmentation during the execution of *EquidistantHistPeaks* as is shown in Fig. 41. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

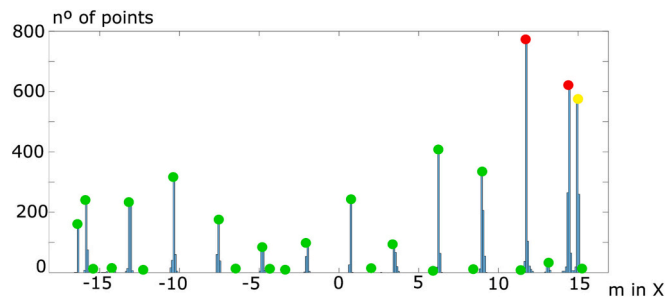


Fig. 41. Histogram of X position of possible strut points. The algorithm *EquidistantHistPeaks* fails in the detection of the X position of the struts because the yellow location is the last one removed. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

to the scanning of truss bridges, it is difficult, and in some cases impossible, to perform scans in such a position that all bridge elements can be visualised without occlusions. Some examples of these occlusions can be observed in Figs. 36, 37, 40, and 42. Analysis of the results showed that most errors occur in areas with more occlusions. This

explains the difference between the results obtained in the two case studies. Case study II has a significantly higher number of false negatives. Most of them are on the interior faces. As shown in Fig. 42, these faces have more occlusions than the interior faces in case study I, as shown in Fig. 18. This difference is due to the scan points used. In case study I, the scan positions are under the bridge; therefore, the board does not occlude the interior faces. This is not possible in case study II because of terrain characteristics. However, it should be noted that despite errors made in areas with occlusions, this algorithm displays a remarkably robust performance in these areas, as shown in Fig. 40, where the algorithm can recognise elements that are not complete in the point cloud. In this sense, the improvement of the results would require not only an improvement in the segmentation algorithm but also in the scanning methods.

The truss bridge topology plays an important role in the segmentation process. The two case study bridges have different topologies. As mentioned previously, the higher incidence of errors in case study II can be explained by occlusions. The topology of this case study with a larger number of members on the interior faces and a half-height board affects the presence of these occlusions. In addition, there are other errors owing to the topology. As shown in Fig. 40, the lateral bottom diagonals are not all the same. Those forming the first, last, and central crosses have different inclinations than the others. The algorithm detects the

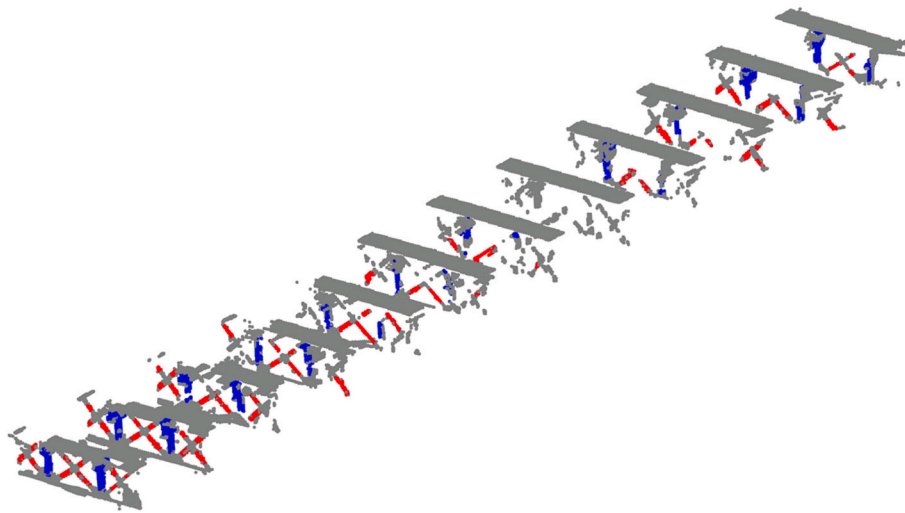


Fig. 42. Interior faces of case study II bridge segmented.

lateral diagonals by calculating their inclinations and assumes that they all have the same angle. This limits the detection of the truss topologies. However, most parts of truss bridges match this requirement, and the algorithm can work in several topologies, as shown in this study.

Finally, it is important to note that despite being a heuristic method with several experimental parameters, only six of them vary between the case studies, and these six are related to the dimensions of the truss elements; thus, they can be measured easily in the point cloud.

## 6. Conclusions

This study presents a methodology for recording a point cloud from truss bridges and automatically segmenting it to obtain information about the elements comprising the truss.

First, data were acquired using a TLS to obtain a point cloud. Second, the point cloud is pre-processed, denoised, and down sampled using a voxelisation method. Third, a segmentation process is applied. The process begins by sectioning the truss into horizontal and vertical faces. The faces were then sequentially segmented, orienting them as vertical faces to apply similar metrics based on the distribution of the neighbourhood at each point. From the results obtained, the interior faces were sectioned and analysed, completing the segmentation process.

A segmented point cloud is obtained using this methodology. The true positives obtained have sufficient information about the item to which they belong, to at least calculate the nodes to which each element is connected.

Two case studies in which this methodology was tested demonstrate good functioning. In the first case study, 272 true positives, 2 false negatives, and 6 false positives are obtained. In the second case study, 154 true positives and 42 false negatives are identified.

Although there is room for improvement, both in terms of metrics and setting more ambitious objectives, such as recognising the sections of the truss elements or completely segmenting them, it is important to emphasise that, to the best of our knowledge, no state-of-the-art truss point cloud segmentation methods have been reported previously. However, the data and codes have been published here to serve as benchmarks for future studies.

There are several interesting future research directions for this study. We have published a paper [31] continuing this work, which demonstrates that this process can be exploited to generate International Foundation Class-compliant models and structural graphs of truss bridges. This continuation is possible because the algorithm presented in this paper is designed as a conservative model, and it does not focus on segmenting all points, but on obtaining groups of points with sufficient

information to represent each element.

## Fundings

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 958171. This work has been partially supported by the Spanish Ministry of Science and Innovation through the PONT3 project Ref. PID2021-124236OB-C33. This document reflects only the views of the authors. Neither the Innovation and Networks Executive Agency (INEA) nor the European Commission is in any way responsible for any use that may be made of the information it contains.

This work has been partially supported by Grant RYC2021-033560-I funded by MCIN/AEI/10.13039/501100011033 and by European Union NextGenerationEU/PRTR.

This work has been partially supported by the University of Vigo through the human resources grant: "Axudas para a contratación de personal investigador predoutoral en formación da Universidade De Vigo 2021" (PREUVIGO-21).

## Declaration of Competing Interest

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

## Data availability

The code and the data are available in a link in the paper

## References

- [1] Vía: Øresund Bridge (912575183), OpenStreetMap. <https://www.openstreetmap.org/way/912575183>, 2023 accessed December 20, 2021.
- [2] M.A. Knudsen, J. Rich, Ex post socio-economic assessment of the Oresund bridge, *Transp. Policy* 27 (2013) 53–65, <https://doi.org/10.1016/J.TRANPOL.2012.12.002>.
- [3] Keeping European Bridges Safe, EU Science Hub. [https://joint-research-centre.ec.europa.eu/jrc-news/keeping-european-bridges-safe-2019-04-05\\_en](https://joint-research-centre.ec.europa.eu/jrc-news/keeping-european-bridges-safe-2019-04-05_en), 2023 accessed July 1, 2022.
- [4] A. Borrmann, M. König, C. Koch, J. Beetz, Building information modeling: Why? What? How?, in: *Building Information Modeling: Technology Foundations and Industry Practice* Springer International Publishing, 2018, pp. 1–24, [https://doi.org/10.1007/978-3-319-92862-3\\_1](https://doi.org/10.1007/978-3-319-92862-3_1).
- [5] Ó. Bouzas, M. Cabaleiro, B. Conde, Y. Cruz, B. Riveiro, Structural health control of historical steel structures using HBIM, *Autom. Constr.* 140 (2022) 104308–104309, <https://doi.org/10.1016/J.AUTCON.2022.104308>.



- [6] F. Shalabi, Y. Turkan, S. Laflamme, BrIM implementation for documentation of bridge condition for inspection, in: Proceedings of the Canadian Society for Civil Engineering 5th International/11th Construction Specialty Conference, University of British Columbia, Vancouver, Canada, 2015, pp. 7–10, <https://doi.org/10.14288/1.0076437>.
- [7] M. Salamak, M. Januszka, BrIM bridge inspections in the context of Industry 4.0 trends, in: Maintenance, Safety, Risk, Management and Life-Cycle Performance of Bridges-Proceedings of the 9th International Conference on Bridge Maintenance, Safety and Management, IABMAS, 2018. [https://www.researchgate.net/profile/Marek-Salamak/publication/327729818\\_BrIM\\_bridge\\_inspections\\_in\\_the\\_context\\_of\\_Industry\\_4\\_0\\_trends/links/5ba127da299b13e603baf8c/BrIM-bridge-inspections-in-the-context-of-Industry-40-trends.pdf](https://www.researchgate.net/profile/Marek-Salamak/publication/327729818_BrIM_bridge_inspections_in_the_context_of_Industry_4_0_trends/links/5ba127da299b13e603baf8c/BrIM-bridge-inspections-in-the-context-of-Industry-40-trends.pdf). accessed December 25, 2022.
- [8] S. Artese, R. Zinno, TLS for dynamic measurement of the elastic line of bridges, Appl. Sci. 10 (2020) 1182, <https://doi.org/10.3390/app10031182>.
- [9] D.G.J. Opoku, S. Perera, R. Osei-Kyei, M. Rashidi, Digital twin application in the construction industry: a literature review, J. Build. Eng. 40 (2021), 102726, <https://doi.org/10.1016/J.JOBE.2021.102726>.
- [10] M. Mohammadi, M. Rashidi, V. Mousavi, A. Karami, Y. Yu, B. Samali, Quality evaluation of digital twins generated based on uav photogrammetry and tls: bridge case study, Remote Sens. 13 (2021) 3499, <https://doi.org/10.3390/RS13173499>.
- [11] M. Rashidi, M. Mohammadi, S.S. Kivi, M.M. Abdolvand, L. Truong-Hong, B. Samali, A decade of modern bridge monitoring using terrestrial laser scanning: review and future directions, Remote Sens. 12 (2020) 3796, <https://doi.org/10.3390/RS12223796>.
- [12] R. Lu, I. Brilakis, C.R. Middleton, Detection of structural components in point clouds of existing RC bridges, computer-aided civil and infrastructure, Engineering. 34 (2019) 191–212, <https://doi.org/10.1111/MICE.12407>.
- [13] 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>.
- [14] B. Riveiro, M.J. DeJong, B. Conde, Automated processing of large point clouds for structural health monitoring of masonry arch bridges, Autom. Constr. 72 (2016) 258–268, <https://doi.org/10.1016/J.AUTCON.2016.02.009>.
- [15] C.R. Qi, H. Su, K. Mo, L.J. Guibas, PointNet: Deep learning on point sets for 3D classification and segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 652–660, <https://doi.org/10.1109/CVPR.2017.16>.
- [16] C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet++: deep hierarchical feature learning on point sets in a metric space, in: Advances in Neural Information Processing Systems 30, Neural Information Processing Systems Foundation, 2017, pp. 5100–5109. <https://arxiv.org/abs/1706.02413v1>. accessed December 21, 2021.
- [17] H. Zhao, L. Jiang, J. Jia, P. Torr, V. Koltun, Point transformer, in: Proceedings of the IEEE International Conference on Computer Vision, 2021, pp. 16239–16248, <https://doi.org/10.1109/ICCV48922.2021.01595>.
- [18] M. Jaderberg, K. Simonyan, A. Zisserman, K. Kavukcuoglu, Spatial Transformer Networks, in: Adv Neural Inf Process Syst, Neural Information Processing Systems Foundation, 2015, pp. 2017–2025. <https://arxiv.org/abs/1506.02025v3>. accessed January 3, 2022.
- [19] H. Kim, J. Yoon, S.-H. Sim, Automated bridge component recognition from point clouds using deep learning, Struct. Control. Health Monit. 27 (2020), <https://doi.org/10.1002/stc.2591>.
- [20] T. Xia, J. Yang, L. Chen, Automated semantic segmentation of bridge point cloud based on local descriptor and machine learning, Autom. Constr. 133 (2022), 103992, <https://doi.org/10.1016/J.AUTCON.2021.103992>.
- [21] G. Andrade, G. Ramos, D. Madeira, R. Sachetto, R. Ferreira, L. Rocha, G-DBSCAN: A GPU accelerated algorithm for density-based clustering, in: Procedia Comput Sci, Elsevier B.V., 2013, pp. 369–378, <https://doi.org/10.1016/j.procs.2013.05.200>.
- [22] F. Hu, J. Zhao, Y. Huang, H. Li, Structure-aware 3D reconstruction for cable-stayed bridges: a learning-based method, computer-aided civil and infrastructure, Engineering 36 (2020) 89–108, <https://doi.org/10.1111/mice.12568>.
- [23] N. Gyetvai, L. Truong-Hong, D.F. Laefer, Laser Scanning-Based Diagnostics In The Structural Assessment Of Historic Wrought Iron Bridges, in: Proceedings of the Institution of Civil Engineers - Engineering History and Heritage, ICE Publishing, 2018, pp. 76–89, <https://doi.org/10.1680/JENHH.17.00018>.
- [24] Z. Shang, Z. Shen, Flight Planning for Survey-Grade 3D Reconstruction of Truss Bridges, Remote Sens. 14 (2022) 3200, <https://doi.org/10.3390/RS14133200>.
- [25] D. Cohen-Or, A. Kaufman, Fundamentals of surface voxelization, Graph. Model. Image Process. 57 (1995) 453–461, <https://doi.org/10.1006/gmip.1995.1039>.
- [26] Y. Wang, L. Cheng, Y. Chen, Y. Wu, M. Li, Building point detection from vehicle-borne LiDAR data based on Voxel Group and horizontal hollow analysis, Remote Sens. 8 (2016) 419, <https://doi.org/10.3390/RS8050419>.
- [27] D. Riaño, E. Meier, B. Allgöwer, E. Chuvieco, S.L. Ustin, Modeling airborne laser scanning data for the spatial generation of critical forest parameters in fire behavior modeling, Remote Sens. Environ. 86 (2003) 177–186, [https://doi.org/10.1016/S0034-4257\(03\)00098-1](https://doi.org/10.1016/S0034-4257(03)00098-1).
- [28] J.F. Weishampel, J.B. Blair, R.G. Knox, R. Dubayah, D.B. Clark, Volumetric lidar return patterns from an old-growth tropical rainforest canopy, Int. J. Remote Sens. 21 (2000) 409–415. <http://www.tandf.co.uk/journals/tf/01431161.html>. accessed November 17, 2021.
- [29] M. Soilán, A. Nóvoa, A. Sánchez-Rodríguez, A. Justo, B. Riveiro, Fully automated methodology for the delineation of railway lanes and the generation of IFC alignment models using 3D point cloud data, Autom. Constr. 126 (2021), 103684, <https://doi.org/10.1016/j.autcon.2021.103684>.
- [30] D. Lamas, M. Soilán, J. Grandío, B. Riveiro, Automatic point cloud semantic segmentation of complex railway environments, Remote Sens. 13 (2021) 2332, <https://doi.org/10.3390/RS13122332>.
- [31] A. Justo, D. Lamas, A. Sánchez-Rodríguez, M. Soilán, B. Riveiro, Generating IFC-compliant models and structural graphs of truss bridges from dense point clouds, Autom. Constr. 149 (2023), 104786, <https://doi.org/10.1016/J.AUTCON.2023.104786>.