Digital twinning of bridges from point cloud data by deep learning and parametric models

M. S. Mafipour, S. Vilgertshofer & A. Borrmann
Chair of Computational Modeling and Simulation
Technical University of Munich, Germany

ABSTRACT: The Digital Twin (DT) of a bridge is a geometric-semantic model that supports and facilitates the operation and maintenance process of the structure. For existing structures, the semantically enriched 3D model of the DT is typically created by processing point cloud data (PCD). Semantic segmentation and parametric modeling are two essential but laborious steps in the digital twinning of bridges. This paper contributes to automating these steps by applying deep learning and metaheuristic algorithms. Semantic features of points are extracted, and a deep learning model is trained. Subsequently, the segmented parts are parametrically modeled by applying a metaheuristic algorithm for model fitting. The presented results show that the DT of bridges can be created with a mean intersection over union (mIoU) of 88.45% and mean accuracy (mAcc) of 95.62% in semantic segmentation, as well as a mean absolute error (MAE) of 4 cm/m in parametric modeling.

1 INTRODUCTION

In most industrialized countries, there is a large stock of aging bridges that require substantial attention for proper maintenance allowing long-term operation. Today, most of the processes involved with inspection, condition rating, and maintenance of these facilities are only loosely supported by digital methods. The "Digital Twin" (DT) concept established originally in the manufacturing industry promises a substantial improvement by providing a coherent digital replica of the physical bridge that is frequently updated to mirror its current condition (Lu et al. 2020, Pan et al. 2019). A major ingredient of DTs in the built environment is coherent semantic-geometric representation of the facility under consideration. Today, the creation of these models is performed mostly manually. As the sheer amount of existing bridges would result in overly high costs and time for a manual DT creation process, it must be (at least partially) automated. This paper contributes to tackling this challenge.

Recent advances in technologies such as photogrammetry and laser scanning have resulted in the fast capturing process of bridges with high measurement accuracy (Adán et al. 2018, Laing et al. 2015, Technion 2015, Rocha et al. 2020). Compared with a visual inspection, these scanning methods can capture the parts which are not easily accessible or even observable (Zhu et al. 2010, Zhu et al. 2010). The resulting point cloud data (PCD) from capturing can visualize the geometric and topological status of an existing bridge and provide a basis to create a DT.

The DT of an existing bridge is formed by a semantically rich 3D model connected with all information from inspections. This information can include cracks, possible areas of defect, and their locations on the body of the structure. Therefore, the DT can support the operation and maintenance process of bridges and reduce the management costs efficiently (Sacks et al. 2016, Sacks et al. 2018). From a structural point of view, it can also be utilized for further analysis of elements and health monitoring of the structure. A DT should be updated in regular intervals to reflect its current conditions. These intervals depend on the type of the product and applications that it can provide (Mafipour et al. 2021). In bridges, these intervals might be longer as the physical features of the structure vary gradually. The DT of a bridge provides a management system that can be used for inspection, assessment, and making more accurate decisions on the possible rehabilitation of the structure.

Despite the prominent role that a DT can play as a support in the maintenance phase of bridges, the manual process of modeling a DT is still labor-intensive, error-prone, and costly. These challenges are mostly due to semantic segmentation and parametric modeling, as necessary steps in digital twinning. To address this issue, these steps need to be automated or at least semi-automated.

2 RELATED RESEARCH

Recently, there have been efforts to automate the modeling process of bridges from PCD (Sacks et al. 2016, Sacks et al. 2018). The proposed methods are mostly based on heuristic algorithms and neural network (NN) models. In general, most heuristic algorithms, in contrast to NN models, do not need a dataset and can directly segment the input PCD. On the other side, NN models are more flexible and less limited to the presumptions that exist in most heuristic algorithms. In what follows, some of the methods employed in the modeling process of bridges are reviewed.

Lu et al. (2019) detected elements in the point cloud of concrete bridges using a top-down approach with geometric cuts. Lee et al. (2020) detected planar faces in the point cloud of bridges and measured the distance between planes to extract the value of parameters for specific types of super-structure. Hu et al. (2021) used a multi-view convolutional neural network (CNN) to extract features from photogrammetry and linked it with a multi-layer perceptron (MLP) to segment the point cloud of bridges. Qin et al. (2021) used the density of the point cloud as a feature in bridges for semantic segmentation and fitted cylindrical and cuboid shapes. Lee et al. (2021) added contextual features by kd-tree and K-nearest neighbors (KNN) to PointNet (Qi et al. 2017) and deep graph-convolutional neural network (DGCNN) (Wang et al. 2019) and improved the performance of the models. Yan & Hajjar (2021) segmented the point cloud of steel bridges based on the connection rules that exist in the bridges. Girardet & Boton (2021) proposed a visual programming approach to foster the parametric modeling of bridges based on technical drawings. Mafipour et al. (2021) provided an approach for parametric modeling of elements that cannot be defined by primitives in point cloud data. Truong-Hong and Lindenbergh (2022) segmented elements in the point cloud of bridges using a voxel growing algorithm. Xia et al. (2022) proposed a local descriptor to calculate the local features of points in a sphere and segmented the point cloud of bridges through a classifier.

3 OVERVIEW ON THE PROPOSED METHOD

This paper proposes an end-to-end method for semantic segmentation and parametric modeling of bridges from PCD. We argue that the use of a parametric model significantly improves the overall quality of the resulting DT in comparison with conventional approaches. In the first part of the paper, the semantic segmentation of the PCD by means of a deep learning model is discussed. To this end, features such as normal vector, 2-D density, and 3-D density of points are calculated. Next, these features, in addition to the *xyz* coordinate of points and RGB channels, are used to train RandLA-Net (Hu et al. 2020). As a training and validation dataset, the PCD of Cambridge (Lu et al. 2019) containing 10 bridges is applied. Four classes of the deck, railing, pier, and background are detected as the important classes in the modeling of bridges. Subsequently, the segmented cloud of a deck is used to create its parametric model. The dummy model of the deck is created based on a series of parameters, and all the human-definable constraints such as symmetry, parallelism, and orthogonality are applied. This initial model is instantiated with random values in ranges inspired by bridge engineering knowledge. Since the parametric model of the deck cannot be defined by a closed-form formulation, particle swarm optimization (PSO) (Kennedy and Eberhart 1995) is employed as a metaheuristic and derivative-free algorithm. The value of parameters is optimized by PSO to fit the dummy model into its corresponding point cloud. This fitted model results in the approximation of the parameters' value representing the point cloud of the element. These values are refined and used to create the 3-D parametric model of the element. All parametric elements are finally assembled into a coherent geometric-semantic model of the bridge forming its digital twin.

4 SEMANTIC SEGMENTATION

Semantic segmentation of a point cloud is the process of assigning individual points to predefined classes. As a result of segmentation, the entire PCD of a bridge is assigned to the bridge parts that represent the elements of interest. In the study presented here, the four classes *Pier*, *Deck*, *Railing* and *Background* are considered, with room for further refinement in the future.

4.1 *Input features*

Input features are the initial inputs that are fed into a neural network (NN). The first layers of a deep network generally process them to extract further features that result in more accurate predictions. In the PCD of bridges, calculating and adding some of these input features can improve the results, thus mixing feature engineering with feature exploration.

4.1.1 *XYZ coordinate and RGB channels*

XYZ coordinate of points and RGB channels are the default features that are used in most NN models with the ability to process points. In bridges, *xyz* of points can represent the local, global, and relative location of elements. RGB can also be a feature emphasizing the parts that have different colors (for ex., the background, the body of the bridge, the road surface, and the railing).

4.1.2 *Normal vectors*

Bridges mostly consist of horizontal and vertical elements. Normal vectors can thus provide a relevant source of information for the network. Normal vectors (n_x, n_y, n_z) are point-level features resulting from the covariance matrix of neighboring points that show the difference in these elements. Horizontal elements generally have higher n_z and lower n_x and n_y than the vertical elements. Since the point cloud of bridges might have rotation around the z-axis, n_x and n_y can have different values in various samples. To make the network invariant to the rotation, the magnitude of the resultant vector from n_x and n_y is calculated and used as a feature:

$$n_{xy} = \sqrt{n_x^2 + n_y^2} \tag{1}$$

4.1.3 *Local variance in the z-direction*

Variance, as a measure of dispersion, demonstrates the distance from the set of points to their mean. Vertical elements in bridges have a higher local variance in the z-direction than horizontal elements. This feature can be calculated locally in a sphere with a predefined radius and computing the value of variance for the z component of points.

4.1.4 *2-D density*

Another feature that can distinguish horizontal and vertical elements is the 2-D density of points. This feature can be calculated by counting the number of points in a circle after projection onto the *xy* plane. Vertical elements have a higher 2-D density than horizontal elements as an impact of overlaying points. This feature, in contrast to *normal vectors*, is not sensitive to noise and can illustrate vertical elements in noisy conditions as well. Note that the resulting PCD of bridges from capturing methods might only have rotation around the z-axis. Thus, this feature can be calculated without any prepossessing on the alignment of bridges.

4.1.5 *3-D density*

Depending on the location of a point and its relative distance to other points in a PCD, it can have different 3-D densities. In the case of bridges, railings and some parts of the background, such as noises, show lower 3-D density. This feature can be calculated by considering a sphere around each point and counting the number of points within the sphere.

4.2 *Dataset*

The bridge dataset of Cambridge (Lu et al. 2019) containing 10 samples is used for training and testing the model. This dataset has been obtained by laser scanning on existing reinforced concrete bridges. All the samples in this dataset are multi-span bridges. In nine out of the 10 samples, abutments cannot be seen, and only two samples have pier caps. Therefore, four classes, including railing, deck, pier, and background, are considered as they exist in all the bridge samples. Leave-one-out cross-validation (LOOCV) is conducted to evaluate the model's performance. 10 folds are considered for training the model. One sample is held out for testing in every fold, and the model is trained on the nine remaining samples.

4.3 *Preprocessing & augmentation*

To compute the input features, the PCD of all the bridges is sub-sampled to voxels with a length of 5 cm by uniform grid sampling. This results in points clouds with one to two million(s) points. All the neighbor-based features are calculated in circles (spheres) with a radius of 20 cm. To equalize the impact of input features on the model, all the features are normalized in the range of zero to one.

To augment the dataset, the PCD of bridges is also translated to the origin of coordinates and rotated by a random angle around the z-axis. They are also jittered with a standard deviation of 1 mm and scaled up or down randomly with a value in the range of 0.9 to 1.1. Due to the different number of points in each class (imbalance dataset), class weights are computed based on the number of points over all the training samples in each class. Subsequently, a higher weight is assigned to the classes with a lower number of points and vice versa. This leads to the equal impact of every class on the loss function of the NN.

4.4 *Deep learning model*

As a deep learning model capable of processing points, RandLA-Net (Hu et al. 2020) is employed. This model can process and learn features of large-scale point clouds. It applies random sampling in subsequent layers of the network to reduce the number of points. Simultaneously, to prevent the loss of key features through sampling, a local feature aggregation module is proposed to progressively increase the receptive field for each point. This module contains a local spatial encoding (LocSE) and an attentive pooling block, as shown in Figure 1. LocSE computes the neighboring points of each point by K-nearest neighbors (KNN) and sends the relative Euclidean distance of the point to its neighbors to a multi-layer perceptron (MLP). Next, the resulting features are concatenated with the input features of the point. In the attentive pooling unit, these features are aggregated through a weighted sum. The weights (scores) of the operation are obtained from a shared MLP to emphasize the more important features. Finally, every two feature aggregation modules are stacked to expand the receptive field of the point.

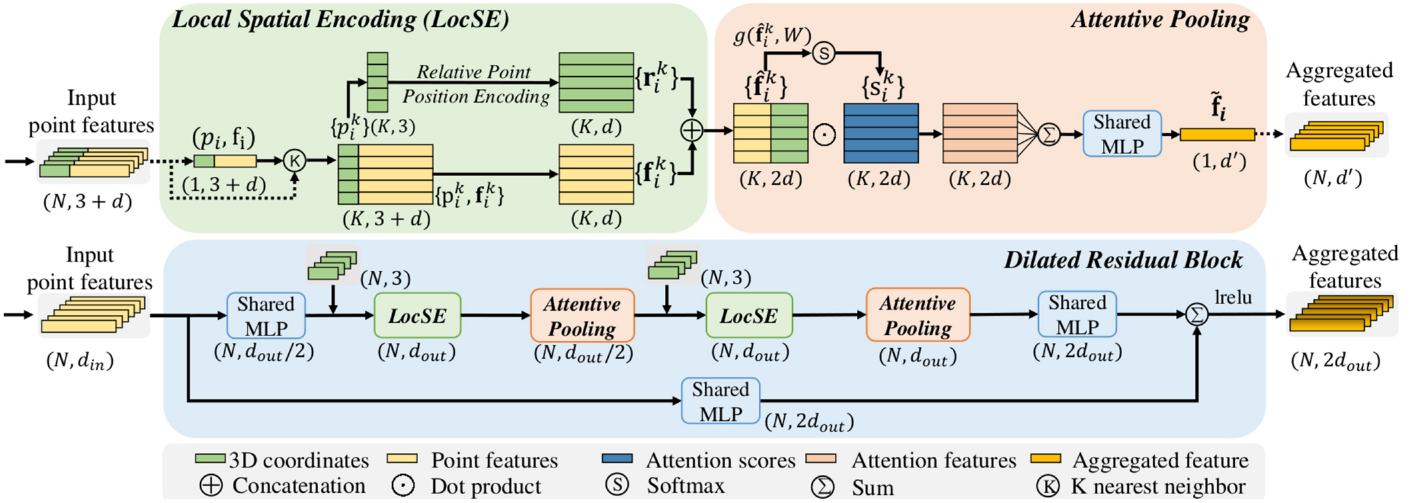


Figure 1: Architecture of RandLA-Net (Hu et al. 2020).

4.5 *Hyperparameters*

RandLA-Net can be configured through a series of hyperparameters. These parameters include the number of sampling layers and their ratio, the number of neighbors required for KNN, and the dimension of features for attentive pooling. It also uses an initial grid sampling which is not necessary but affects the performance of the model. The remaining parameters, such as batch size and learning rate, mostly correspond to the training phase of the model. All these parameters are obtained through a trial and error process, and the values leading to the best results are reported. For the dataset of bridges, four sampling layers with a ratio of 1/4 are applied so that only 25% of points in each layer are retained. 16 neighbors are considered for KNN, and 8 neurons for processing features in the attentive pooling are selected. A batch size of 3 and a learning rate of 0.01 are applied for training the model.

4.6 *Statistical metrics*

Two statistical metrics of accuracy (Acc) and intersection over union (IoU) are used to evaluate the performance of the model in the training and testing phases. These metrics are calculated from the confusion matrix of the prediction results and are defined as below:

$$Acc = \frac{TP}{TP + FN} \tag{2}$$

$$IoU = \frac{TP}{TP + FN + FP} \tag{3}$$

where *TP*, *FN*, and *FP* are true positive, false negative, and false positive, respectively.

Since these metrics are calculated for each class, their mean values over classes, i.e., mean accuracy (mAcc) and mean IoU (mIoU), are reported as statistical indices of the model.

4.7 *Results*

RandLA-Net with the extracted input features was trained for 512 epochs on the folds obtained from LOOCV. Every fold contained nine samples of bridges for training and one unseen sample for testing. Figure 2 shows the resulting learning curves of the model, including the values of loss, mIoU, and mAcc on a typical fold in the training phase. All the trained models resulted in almost the same learning diagrams in the training phase with a mAcc and mIoU around 98% and 96%, respectively.

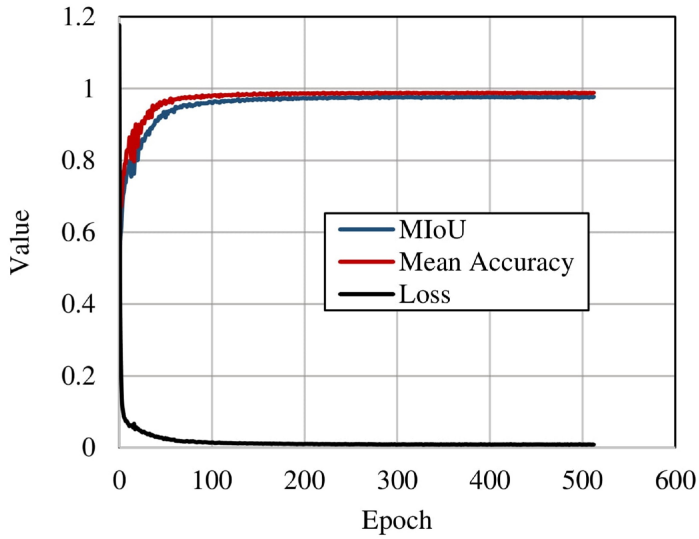
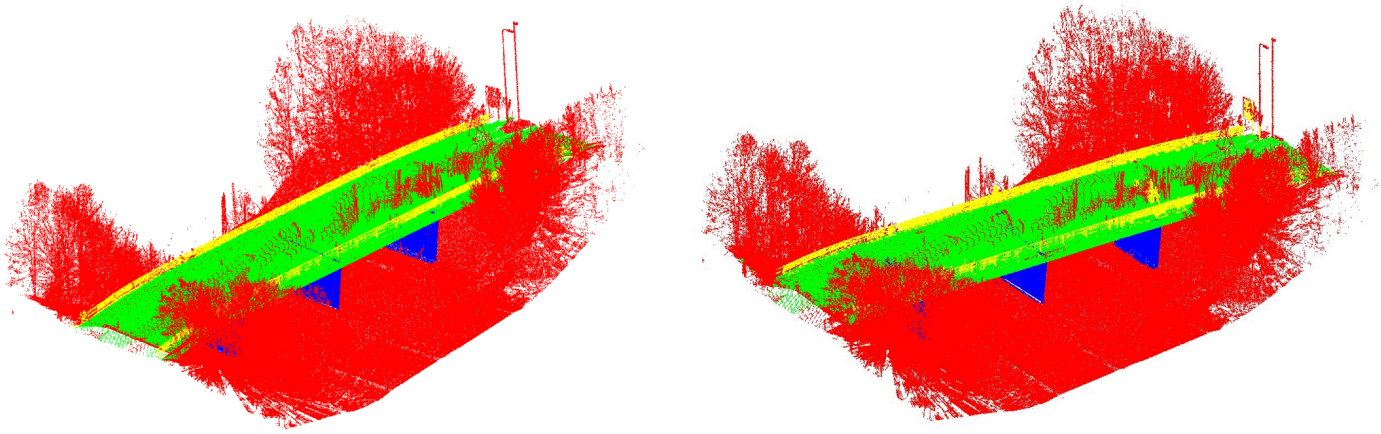


Figure 2: Learning diagrams of the model in the training phase.

After training, the performance of the models was tested on one unseen sample. Table 1 illustrates the results in the testing phase. As can be seen, most of the models have resulted in a mAcc and mIoU more than 94% and 89%, respectively. Among these samples, *Bridge1*, *Bridge7*, and *Bridge9* have shown the lowest mIoU. This can be due to the higher difference in the PCD between these samples and the training samples. *Bridge1* and *Bridge9* are the two only samples that have pier cap and this element cannot be seen

Table 1: The result of K-fold cross-validation (%)

Test sample	Pier		Deck		Railing		Background		mIoU	mAcc
	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc		
Bridge1	63.02	65.73	85.77	98.61	82.60	91.73	96.70	96.85	82.02	88.23
Bridge2	94.35	97.36	90.82	91.83	80.89	89.72	97.70	99.73	90.94	94.66
Bridge3	96.47	98.48	95.33	97.54	83.18	98.31	98.45	98.75	93.36	98.27
Bridge4	94.33	99.69	95.08	96.83	70.71	95.76	96.82	98.17	89.24	97.61
Bridge5	95.91	98.84	93.52	96.52	70.38	99.03	98.09	98.18	89.47	98.14
Bridge6	96.05	99.81	94.22	95.62	75.51	99.63	98.35	98.90	91.03	98.49
Bridge7	85.14	99.74	88.01	88.63	71.47	99.02	83.99	87.38	82.15	93.69
Bridge8	95.87	99.15	94.43	95.19	73.14	97.03	98.39	99.14	90.46	97.63
Bridge9	95.59	99.45	91.35	93.45	82.07	86.76	95.03	98.70	91.01	94.59
Bridge10	83.81	99.15	89.18	98.44	72.32	87.45	93.94	94.33	84.81	94.84



Ground truth Prediction of the model

Figure 3: Comparison of the ground truth with the prediction of the model. Pier: blue; background: red; deck: green; railing: yellow

in other samples. Also, *Bridge7* is the only sample in which there are two large abutments at the ends of the bridge deck. Table 2 shows the mean value of IoU and Acc over each class in the testing phase of models. Considering the values of mIoU and mAcc, the class of *railing* has the lowest accuracy. This can also be due to the noise that is generated around railings during capturing. This noise can result in the incorrect calculation of some features such as normals that are important in the segmentation of these elements.

Figure 3 depicts the resulted labels from the test sample *Bridge2* and illustrates a visual comparison between the predicted labels and the ground truth. As can be seen, most of the elements except some parts of the railing that are close to the deck have been labeled correctly. After railing, piers and deck seem to be the hardest parts for the network to detect, especially at locations where these elements are connected. Averaging the values of accuracy in the testing phase of all the bridges, semantic segmentation of bridges can be conducted with an mIoU of 88.45% and mAcc of 95.62%.

Table 2: mAcc and mIoU over classes (%)

Class	mIoU	mAcc
Pier	90.05	95.74
Deck	91.77	95.27
Railing	76.23	94.44
Background	95.74	97.01

5 PARAMETRIC MODELING

Parametric modeling is a computer-aided design (CAD) approach that results in a dynamic model with the capability of changing shape. A parametric model includes a finite number of parameters through which it can be steered. The model also contains a number of constraints that preserve the overall shape of the element while being updated.

To create the parametric model of elements in bridges, a model-based cloud fitting approach is proposed. In contrast to the conventional methods that are mostly limited to either surfaces or simple 3D primitives, this technique can be used for non-primitive elements that exist in bridges commonly. It can also result in a geometric-semantic model with higher quality and coherence than conventional approaches.

5.1 *Cloud fitting*

Semantic segmentation of bridges results in the separated point cloud of elements and labels representing the class (bridge part) the points belong to. To fit a model to these clouds, the prototype model of the element is created. This prototype model is created based on the obtained type from semantic segmentation and preknowledge of the geometric features of the element. To create the prototype model, a 2-D profile is programmed as shown in Figure 4. This profile is defined by a set of parameters that can form a closed polygon. These parameters include the coordinate of origin, length of edges, or angles. Constraints such as parallelism, symmetry, and orthogonality are programmed as well and applied to the edges implicitly. To fit a prototype model, the minimum Euclidean dis-

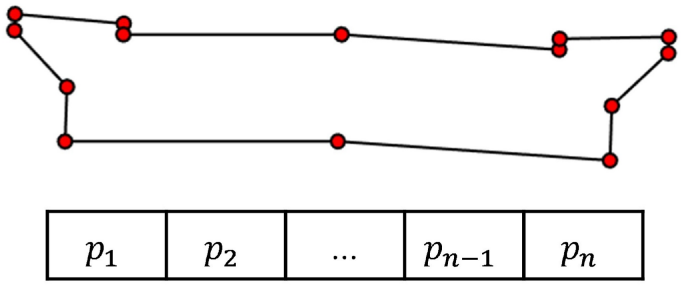


Figure 4: The 2-D dummy model of a deck with its parameters.

tance of points to the vertices and edges of the profile is calculated. This overall distance is then defined as the mean absolute error (MAE) or root mean squared error (RMSE) and minimized to fit the profile into the cloud (Mafipour et al. 2021):

$$F(p_1, p_2, ..., p_n) = \frac{\sum_{i=1}^{N} min(d_{ij}(b_i, P(v_j, e_j)))}{N} \quad (4)$$

where d_{ij} is the distance of the profile P with vertices v_j and edges e_j to the boundary points b_i. N is the total number of points as well.

As shown in Equation 4, the parameters set $\{p_1, p_2, ..., p_n\}$ that form the prototype model cannot be directly seen in the defined cost function. As a result, conventional gradient-based algorithms cannot simply solve the problem as they need the partial derivatives of the cost function with respect to the parameters. To address this issue, particle swarm optimization (PSO), as a well-established metaheuristic algorithm, is employed (Kennedy and Eberhart 1995). Metaheuristic algorithms, in contrast to gradient-based algorithms, are not dependent on the derivatives of the fitness function and can optimize minimization or maximization problems.

5.2 *Particle swarm optimization (PSO)*

Particle swarm optimization (PSO) is an optimization technique inspired by the immigration pattern of birds (Kennedy and Eberhart 1995). In this algorithm, a swarm of particles (solutions) is randomly distributed in the *n*-dimensional space of the problem. Each solution is evaluated using the fitness function of the problem. Next, the location of every particle is updated based on its local best position and the global best position of the swarm. This process is repeated until the applied stopping criteria to the problem is satisfied.

For model fitting of elements in bridges, every solution of PSO simulates a parametric profile, with each parameter providing a dimension in the solution space. A swarm of particles is also a group of these randomly simulated profiles. The parameters of each profile are encoded as a solution in PSO. These profiles are adjusted by PSO and fitted into the cloud. As a result, the parameters after optimization acquire the actual values that the point cloud is representing. Due to the parametric design of each profile, the number of parameters is finite and logically compatible with the BIM-authoring systems. Therefore, the value of these parameters after optimization can be imported to the parametric model of the element to create a volumetric 3-D model.

5.3 *Results*

For parametric modeling, the detected deck in the testing phase of the *Bridge2* was considered. Decks are geometrically complicated elements for modeling in bridges. Most of the existing decks cannot be defined by primitives. A deck might have horizontal and vertical curvature depending on the roadway and topography of the region. Therefore, the alignment of the bridge is to be recognized first. The underlying assumption is that the orientation of the longer axis of the point cloud follows the alignment. For this purpose, the segmented points of the deck are projected onto the *xy* plane. These 2-D points are then subsampled by uniform grid sampling to make them even. The alignment of most of the conventional bridges can be approximated by a polynomial with a degree of two. Therefore, this polynomial is fitted to the projected points, thus, representing the alignment of the deck (bridge).

To extract the value of profile parameters, the points of the deck are segmented into intervals along the length of the polynomial. Next, the points of each segment are rotated around the z-axis and projected onto the *yz* plane. This step can be performed by using the vectors connecting every two sequential segments

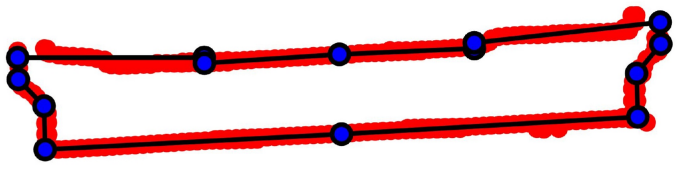


Figure 5: The fitted prototype model into the point cloud by PSO.

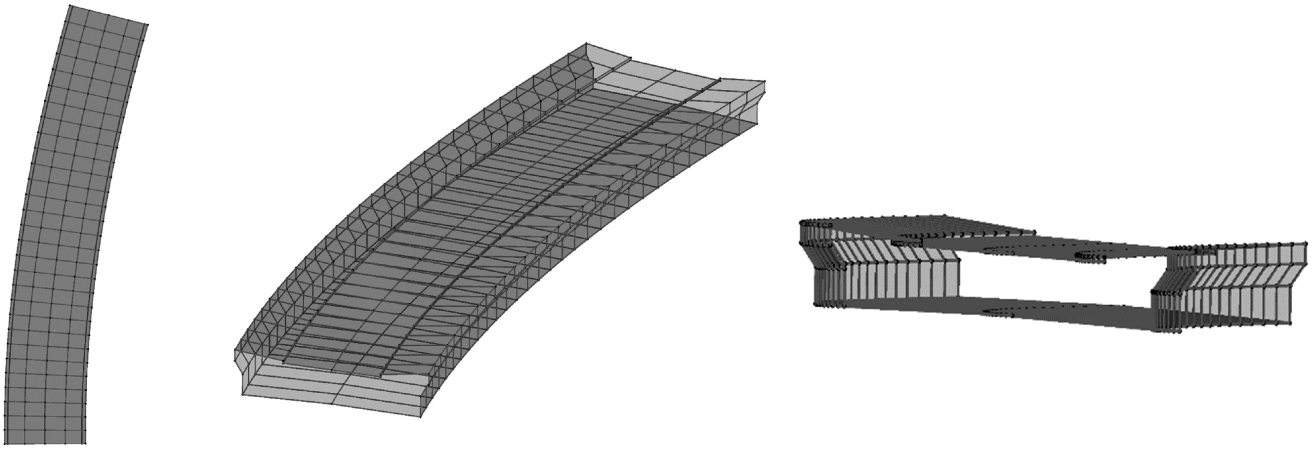


Figure 6: Fitted model the the segmented point cloud of the deck.

along the length of the polynomial. The resulting projected segments can finally be used for applying PSO. Based on the type of the deck, a hollow deck section, as shown in Figure 4, is used. For model fitting, the deck of the bridge is segmented in intervals of 1 m. PSO is set up with 35 particles and coefficient factors of c_1 , $c_2 = 2$ as well as a damping factor of 0.99. Each segment of the deck is fitted separately for 200 iterations. Figure 5 shows one of the fitted segments along the length of the bridge.

Considering the noises that generally exist, the optimization algorithm showed an MAE of 4 cm (or 4 cm/m). After model fitting, the value of parameters for each segment is saved. Since each segment results in almost different values, the value of parameters needs to be refined. To this end, the outlier values are excluded, and the average values for the remaining values are calculated. This process results in a set of parameters that represent all the fitted segments together. Finally, the 3-D model of the deck is created by the sweep path (polynomial) and the profile, as shown in Figure 6.

6 CONCLUSION

This paper presents an end-to-end automated process to create the parametric model of bridges from point cloud data (PCD). In the first part of the paper, features of points in the point cloud of bridges are extracted. These features include normal vectors, variance in the z-direction, 2-D density, and 3-D density. Features such as 3-D density and RGB are effective in the segmentation of background from the structure, while variance in the z-direction, 2-D density, and normal vectors can assist in the detection of horizontal and vertical elements. Considering bridges that consists of horizontal and vertical faces as well as different densities and colors, these features can be efficient. The results of k-fold cross validation shows that the point cloud of bridges can be segmented properly with a mean intersection over union (mIoU) of 88.45% and a mean accuracy (mAcc) of 95.62%.

In the second part of the paper, a methodology for parametric modeling of bridge elements has been presented. In bridges, most of the elements cannot be defined as primitives and have more complicated shapes. This results in the lack of closed-form formulations for model-fitting. To overcome this problem, the parametric profile of the elements is created and all the human definable constraints such as symmetry, parallelism, and orthogonality are applied. These profiles are adjusted and optimized to be fitted into the point cloud by an optimization algorithm. Since particle swarm optimization (PSO) is a derivative-free algorithm, it can be used for the model fitting process. All the desired parameters can be encoded in PSO as a solution and adjusted by the algorithm. The results of the model fitting to the different segments of a deck along its length show that the value of parameters can be extracted appropriately with a mean absolute error (MAE) of 4 cm/m. Considering the results of semantic segmentation and parametric modeling, the scan-to-BIM process of existing bridges can be automated to a large extent. However, the proposed algorithms still have some limitations. In addition to the studied elements (classes) in the paper, the two classes *Abutment* and *Pier Cap* need to be detected in bridges as well. The segmentation of these elements requires a dataset in which they can be observed and annotated adequately. In addition, a larger dataset can provide the opportunity to conduct a thorough test on the point cloud of more complex bridges.

ACKNOWLEDGMENT

The research presented has been performed in the frame of the TwinGen project funded by the German Ministry for Digital and Transport (BMDV).

REFERENCES

Adán, A., B. Quintana, S. A. Prieto, & F. Bosché (2018). Scan-to-bim for 'secondary' building components. *Advanced Engineering Informatics 37*, 119–138.

Girardet, A. & C. Boton (2021). A parametric bim approach to foster bridge project design and analysis. *Automation in Construction 126*, 103679.

Hu, F., J. Zhao, Y. Huang, & H. Li (2021). Structure-aware 3d reconstruction for cable-stayed bridges: A learning-based method. *Computer-Aided Civil and Infrastructure Engineering 36*(1), 89–108.

Hu, Q., B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, & A. Markham (2020). Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11108–11117.

Kennedy, J. & R. Eberhart (1995). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, Volume 4, pp. 1942–1948. IEEE.

Laing, R., M. Leon, J. Isaacs, & D. Georgiev (2015). Scan to bim: the development of a clear workflow for the incorporation of point clouds within a bim environment. *WIT Transactions on The Built Environment 149*, 279–289.

Lee, J. H., J. J. Park, & H. Yoon (2020). Automatic bridge design parameter extraction for scan-to-bim. *Applied Sciences 10*(20), 7346.

Lee, J. S., J. Park, & Y.-M. Ryu (2021). Semantic segmentation of bridge components based on hierarchical point cloud model. *Automation in Construction 130*, 103847.

Lu, Q., L. Chen, S. Li, & M. Pitt (2020). Semi-automatic geometric digital twinning for existing buildings based on images and cad drawings. *Automation in Construction 115*, 103183.

Lu, R., I. Brilakis, & C. R. Middleton (2019). Detection of structural components in point clouds of existing rc bridges. *Computer-Aided Civil and Infrastructure Engineering 34*(3), 191–212.

Mafipour, M. S., S. Vilgertshofer, & A. Borrmann (2021). Deriving digital twin models of existing bridges from point cloud data using parametric models and metaheuristic algorithms. In *Proc. of the EG-ICE Conference 2021*.

Pan, Y., A. Borrmann, H.-G. Mayer, F. Rhein, C. Vos, E. Pettinato, & S. Wagner (2019). Built environment digital twinning. Report, Technical University of Munich.

Qi, C. R., H. Su, K. Mo, & L. J. Guibas (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660.

Qin, G., Y. Zhou, K. Hu, D. Han, & C. Ying (2021). Automated reconstruction of parametric bim for bridge based on terrestrial laser scanning data. *Advances in Civil Engineering 2021*, 17.

Rocha, G., L. Mateus, J. Fernández, & V. Ferreira (2020). A scan-to-bim methodology applied to heritage buildings. *Heritage 3*(1), 47–67.

Sacks, R., A. Kedar, A. Borrmann, & L. Ma (2016). *SeeBridge information delivery manual (IDM) for next generation bridge inspection*, Book section 1, pp. 826–834. ISARC.

Sacks, R., A. Kedar, A. Borrmann, L. Ma, I. Brilakis, P. Hüthwohl, S. Daum, U. Kattel, R. Yosef, & T. Liebich (2018). Seebridge as next generation bridge inspection: overview, information delivery manual and model view definition. *Automation in Construction 90*, 134–145.

Technion (2015). Seebridge—semantic enrichment engine for bridges. Report, Technion.

Truong-Hong, L. & R. Lindenbergh (2022). Automatically extracting surfaces of reinforced concrete bridges from terrestrial laser scanning point clouds. *Automation in Construction 135*, 104127.

Wang, Y., Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, & J. M. Solomon (2019). Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG) 38*, 1–12.

Xia, T., J. Yang, & L. Chen (2022). Automated semantic segmentation of bridge point cloud based on local descriptor and machine learning. *Automation in Construction 133*, 103992.

Yan, Y. & J. F. Hajjar (2021). Automated extraction of structural elements in steel girder bridges from laser point clouds. *Automation in Construction 125*, 103582.

Zhu, Z., S. German, & I. Brilakis (2010). Detection of large-scale concrete columns for automated bridge inspection. *Automation in construction 19*(8), 1047–1055.