Deriving Digital Twin Models of Existing Bridges from Point Cloud Data Using Parametric Models and Metaheuristic Algorithms

M. Saeed Mafipour, Simon Vilgertshofer, André Borrmann
Chair of Computational Modeling and Simulation, Technical University of Munich, Germany
m.saeed.mafipour@tum.de

Abstract. In building information modeling (BIM), a digital twin (DT) is a model that represents the current status of an existing structure; thus, facilitating the operation and management process. Due to higher measurement speed and accuracy, laser scanning and photogrammetry are generally employed, resulting in point cloud data (PCD). Today, the required volumetric models are created in a laborious and costly manual process from PCD. This paper aims to automate this process by applying metaheuristic optimization algorithms to fit highly parametric BIM models of bridges into given point clouds. For this purpose, parametric base models of elements are created and instantiated by adjusting their parameters' value using metaheuristic algorithms. This optimization process leads to extracting the parameters for a model from PCD and creating 3-D volumetric shapes. The paper's results show that metaheuristic algorithms can be successfully used for parametric modeling even in point clouds with occlusion and clutter.

1. Introduction

Building information modeling (BIM) is an efficient tool for supporting the design and construction of buildings and infrastructure facilities. BIM can also assist in the operation and maintenance process. *As-is* BIM models represent the digital replica of an existing facility, such as a bridge, and provide an appropriate basis for inspection, condition assessment, and repair planning (Sacks et al., 2018). They also provide an integrated and single unit in which all the gathered information from the construction site can be imported. The main advantages of a digital as-is model are the possibility of accessing and querying structured data and the visualization of information.

Most recently, the concept of as-is BIM has been extended to digital twin (DT) (Pan et al., 2019; Lu et al., 2020). A DT is updated frequently, thus keeping the digital replica consistent with the physical reality. However, the frequency of these updates depends on the product type, its dynamics, and the model's purpose. While the DT of a jet engine is updated in minute intervals, it is suitable to update the DT yearly in bridge maintenance management. However, a significant challenge is that the vast majority of existing bridges were constructed decades ago, which means DT models must be created from the existing asset as well.

Laser scanning and photogrammetry are two of the best-known methods to capture the geometry of an existing facility (Bosché et al., 2015; Laing et al., 2015; Technion, 2015; Adán et al., 2018; Rocha et al., 2020). The output of these techniques is point cloud data (PCD). Compared with a visual inspection, PCD is provided in a lower time and has higher measurement accuracy (Zhu et al. 2010). However, DT modeling based on PCD is laborious and error-prone. In current practice, these models are created manually, which in turn, increases the duration and costs. Hence, infrastructure authorities mostly do not undertake high costs and potential risks of DT models and still prefer the old rating system to manage structures (Zhu et al. 2011). To utilize the benefits of DT models and reduce the modeling costs, the digital twinning process needs to be automated. Recently, several attempts have been made towards

this goal (Sacks et al., 2016; Sacks et al., 2018), which mostly follow a bottom-up approach, which has limitations, especially in point clouds with occlusion and clutter.

In this paper, we propose a method based on metaheuristic algorithms to automate the creation of parametric BIM models. We use a top-down approach for parametric modeling of bridges from PCD and combine it with a bottom-up approach by instantiating parametric profiles of bridge elements. These profiles are created based on pre-knowledge about the existing elements in a typical bridge. Hence, the profiles comprise all the human-definable features such as parallelism, symmetricity, and orthogonality. Since the scope of the paper is on parametric modeling, we use element-wisely segmented point clouds. Also, it is assumed that elements can be defined by an *extrude* function. To extract the parameters' value, the required cross-section or face for the *extrude* function is recognized and then be used as an input for updating the parameters' value of the corresponding profile. Since closed-form formulations cannot describe these profiles, metaheuristic algorithms are applied. Finally, all the extracted parameters are used to create the parametric model of the elements. The workflow of the proposed approach can be seen in Figure 1.

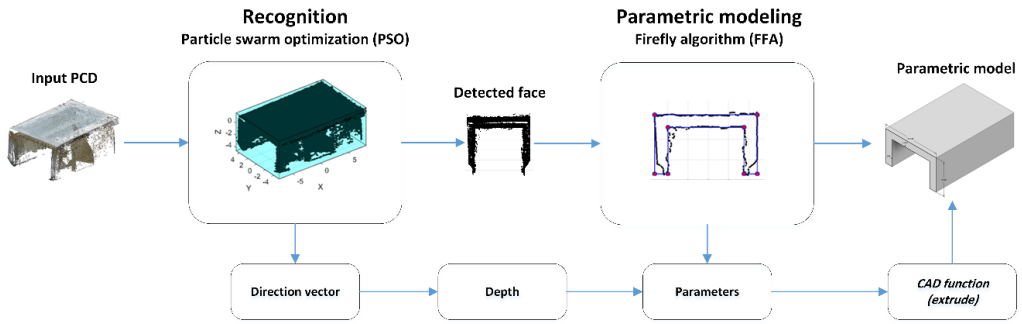


Figure 1: The proposed pipeline of parametric modeling

2. Related research

Bottom-up and top-down are the major approaches for detecting structural elements and modeling based on PCD. The bottom-up methods start from the low-level features to generate a complex system at successively higher levels. Walsh et al. (2013) extracted sharp features of points and used a region-growing algorithm to segment planar faces of bridge elements. Next, surfaces were fitted by the least square algorithm. Zhang et al. (2015) determined the local features of points and clustered them based on the existing linear relationships and finally extracted the planar faces of elements in bridges by singular value decomposition (SVD). Yan et al. (2017) used principal component analysis (PCA) to recognize the endpoints of elements and then applied a voxelization process to modify the real boundaries of bridges to generate a mesh. The bottom-up approach provides an efficient tool for modeling elements. However, created models are vulnerable to occlusion and do not mostly provide a meaningful parametric model in the end.

In contrast to the bottom-up, the top-down methods start from an abstract model and decompose a complex system to the subordinate models. Lu et al. (2019) used a top-down approach for detecting elements in the point cloud of RC bridges and represented the geometry of the bridge by the alpha-concave hull. Qin et al. (2021) also considered a top-down approach for detecting elements in bridges based on the density of points and employed a bottom-up method for parametric modeling of cylindrical and cuboid shapes. Kwon et al. (2004) introduced a fast and accurate local spatial modeling algorithm to fit planes, cuboids, and cylinders to sparse PCD, assuming that the construction site can be modeled by these primitives. Song and Jüttler (2009)

improved the performance of implicit modeling by adding sharp features to the models. Cao and Wang (2019) used cuboids and graph-cut energy minimization algorithms for model fitting to unstructured PCD. The top-down methods can provide a completely human-understandable model; however, they have been mostly limited to primitives as they can be mathematically defined in closed-form formulations.

2.1 Overview of metaheuristic algorithms

Metaheuristic algorithms are a sub-branch of optimization algorithms and artificial intelligence. These algorithms have been inspired mainly by natural, biological, and social systems of animals and humans. In contrast to most optimization algorithms, metaheuristic algorithms do not need the closed-form formulation of the loss function. Hence, they can be adequately used for expressing a parametric instance with no closed-form formulation.

Particle swarm optimization (PSO)

PSO, as a metaheuristic algorithm, was proposed by (Kennedy and Eberhart, 1997). This swarm intelligence algorithm has been inspired by the social behavior of birds and herds of fish. A population of random solutions is firstly initialized in PSO, called a swarm of particles. Based on a fitness function, the quality of solutions is assessed. Next, the position of each particle is updated by the following formulas:

$$V_i^{k+1} = w.V_i^k + c_1.r_1.(P_{best,i}^k - p_i^k)/\Delta t + c_2.r_2.(G_{best}^k - p_i^k)/\Delta t \quad (1)$$

$$p_i^{k+1} = p_i^k + V_i^{k+1}.\Delta t \quad (2)$$

where p_i is the position of the i^{th} particle, V_i is the velocity vector of the i^{th} particle, $p^k{}_{best,\,i}$ is the best position of the i^{th} particle over its history up to iteration k, $G^k{}_{best}$ is the position of the best particle in the swarm by up to iteration k, c_1 is the cognitive parameter, c_2 is a social parameter, r_1 and r_2 are independent random numbers uniformly distributed between 0 and 1, w is the inertial weight, and Δt is the time interval which is considered equal to 1.

Firefly algorithm (FFA)

FFA is another metaheuristic algorithm that was proposed by (Yang, 2008). This algorithm has been inspired by fireflies' flashing patterns to attract their partners, communicate, and show risk warnings. Every firefly is assumed unisexual in FFA whose attractiveness is proportional to its brightness. This algorithm is based on three parameters, including attractiveness, randomization, and absorption. The position of every firefly is formulated as below:

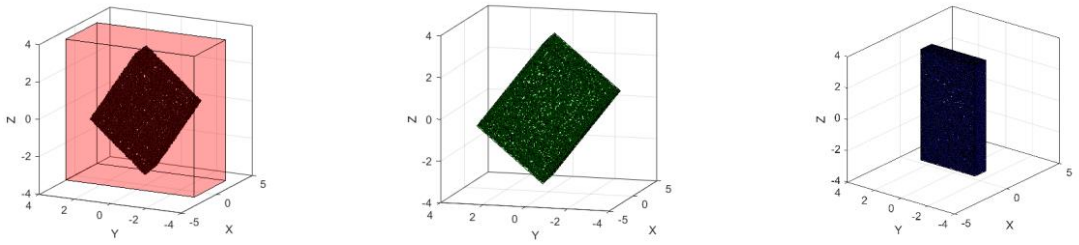
$$x_i^{t+1} = x_i^t + \beta_0 e_i^{-\gamma r_{ij}^2}(x_j^t - x_i^t) + \alpha_t \varepsilon_i^t \quad (3)$$

where x_i is the position of a firefly at the iteration t, $\beta_0 > 0$ is the attractiveness at the distance zero ($r_{ij} = 0$), γ is the absorption coefficient that controls the visibility of fireflies, ε_i is a vector with random numbers, and α_t is the mutation coefficient.

3. Method for model-to-cloud fitting

To extract the parameters' value of an element from its corresponding point cloud, the cross-section or face of interest should be recognized first. In this paper, the required face for the *extrude* function is only detected since most of the elements in bridges, including piers, wing walls, and direct decks, can be defined by this function. To this end, we evaluate all the faces of the element by a bounding box. Figure 2(a) illustrates the axis-aligned bounding box (AABB)

of a point cloud that is not aligned coordinate axes. As can be seen, the lack of alignment in the point cloud has resulted in an AABB that is not the minimal bounding box (MBB) simultaneously (Figure 2(b)). Based on this observation, if the AABB of a point cloud is its MBB at the same time, the point cloud is thus aligned coordinate axes. We denote this resulting bounding box as Minimal Axis Aligned Bounding Box (MAABB) that aligns point cloud in the direction of coordinate axes (see Figure 2(c)). A MAABB is computed the same as an AABB, however, after applying a transformation to the point cloud.



(a) (b) (c)

Figure 2: Different types of bounding box: (a) AABB; (b) MBB; (c) MAABB

To determine this transformation, an optimization problem is defined. As the first step, the point cloud is translated to the origin of the coordinate system. Next, it is transformed using the general form of the rotation matrix in 3-D space. This rotation matrix can be computed by the multiplication of rotation matrices around *x*, *y*, and *z* axes with the angles α, β, and γ, respectively. For every value of α, β, and γ, a new rotation matrix can be obtained, and a volume for the AABB can thus be calculated. Hence, the fitness function of the optimization problem can be defined as below:

$$\textit{To Minimize: } V(\alpha,\beta,\gamma) = l \times w \times h \qquad \textit{Subjected to: } -\pi \le \alpha,\beta,\gamma \le \pi \tag{4}$$

where *V* is the AABB volume after transformation, *l*, *w*, and *h* are also the dimensions of the bounding box.

All the available real (continuous) metaheuristic algorithms can solve this optimization problem. In this paper, PSO is used as it is simple in coding and results in faster convergence.

3.1 Extrude function

Two parameters are required to extrude a 2D sketch: the direction vector and thickness (depth). Using MAABB, these parameters are simply determined. Figure 3 shows an element created by the extrude operation. As can be seen, the shape's projection (shadow) is a rectangle in all the side views, except for the face of interest (the extrusion base plane). This feature is seen in any shape created by extrusion. Therefore, the face with the lowest similarity to a rectangle is selected as the basis of the extrusion. Subsequently, the vector perpendicular to this face is the direction vector, and the dimension along this vector is the thickness.

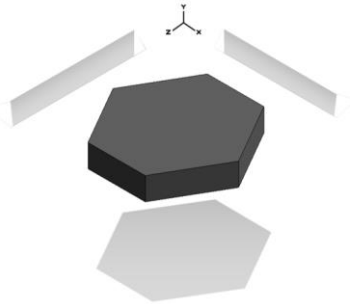


Figure 3: An arbitrary element created by extrude function

To determine the similarity between the faces of a point cloud and a rectangle, the factor of area ratio is defined. This factor is the ratio of the covered area by points and the faces area of the MAABB. The former can be estimated by creating the alpha complex of points with a critical value of alpha that leads to a single region, i.e., a_{xy}, a_{xz}, and a_{yz}, and the latter is simply computed using the dimensions of the MAABB as follows:

$$A_{xy} = L_x L_y , \quad A_{xz} = L_x L_z , \quad A_{yz} = L_y L_z \tag{5}$$

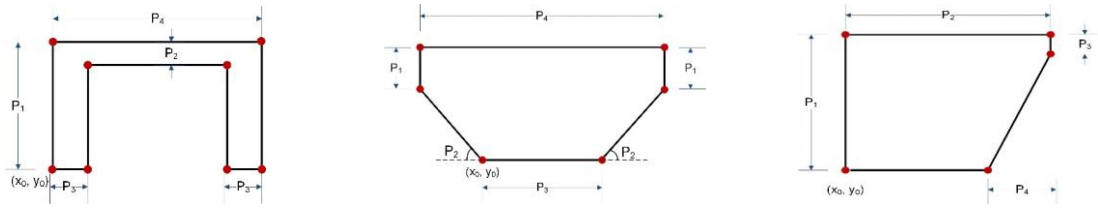
Considering the calculated values of the area, one area ratio for each direction (x, y, z) can be obtained as below:

$$r_x = a_{yz} / A_{yz} , \quad r_y = a_{yz} / A_{yz} , \quad r_z = a_{xy} / A_{xy} \tag{6}$$

The minimum area ratio shows the direction of extrusion in MAABB.

3.2 Parametric modeling

The methodology described in the previous section can detect the cross-section of any element that can be modeled by the *extrude* function. This element can be a wing wall, a straight deck, or an abutment in a typical bridge. To extract the parameters' value of elements, the corresponding profiles of the elements can be created based on pre-knowledge, as shown in Figure 4. Although these profiles cannot be expressed by closed-form formulations, they all can be defined by an origin (x_0, y_0) and a set of parameters {p_1, p_2, …, p_k}.



(a) (b) (c)

Figure 4: Parametric profiles: (a) Wing walls; (b) Deck; (c) Abutment

Adjusting the origin and value of the parameters leads to new geometries. Hence, if these profiles are optimized and become closer to the existing points, the obtained parameters at the end of the optimization process will be the actual parameters of the profile. For this purpose, we use metaheuristic algorithms and encode every solution as shown in Figure 5.

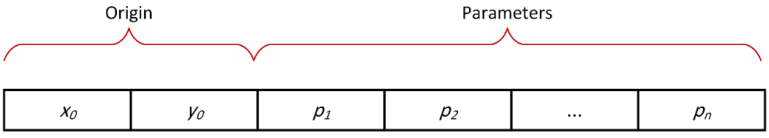


Figure 5 Encoding a profile as a solution in a metaheuristic algorithm

Given a set of points $\mathcal{B} = \{b_i = (x_i, y_i), i = 1, 2, \ldots, N\}$ and the profile $F(v_j, e_j)$ with vertices v, edges e, and parameters set $P = \{x_0, y_0, p_1, p_2, \ldots, p_k\}$, the fitness function of the optimization problem can be defined as the root min squared error (*RMSE*) of the minimum distance between the profile and points as below:

$$\textit{To minimize: } D(P) = \sqrt{\frac{\sum_{i=1}^{N} \min(d_{ij}(b_i, F(v_j, e_j)))^2}{N}} \quad j = 1, 2, \ldots, M \quad \& \tag{7}$$

$$(v_j, e_j) = f(x_0, y_0, p_1, p_2, \ldots, p_k)$$

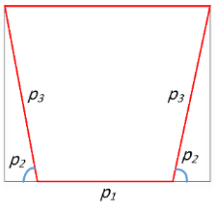
where d_{ij} (b_i, $F(v_j, e_j)$) is the distance of the i^{th} point to the j^{th} vertex v or edge e. N is the number of points, M is the number of vertices or edges, and k is the number of parameters.

The range of parameters needs to be defined as well to solve this optimization problem. These ranges can be estimated based on engineering knowledge or can be provided by external resources. Note that the exact ranges are not required, and they should be defined such that the profile can keep its form during the optimization process. However, to make the profiles adaptive, a simple method for estimating these ranges is proposed. For this purpose, all the points are normalized in the range of [-1, 1] using the following formula:

$$(x_i, y_i) = \frac{(x_{io}, y_{io}) - (x_{\min}, y_{\min})}{(x_{\max}, y_{\max}) - (x_{\min}, y_{\min})} \times 2 - 1 \tag{8}$$

where (x_i , y_i) and (x_{io}, y_{io}) are the coordinates of points after and before normalization, respectively. (x_{min} , y_{min}) and (x_{max} , y_{max}) are also the minimum and maximum of points.

After this process, all the points will be mapped in a square bounding box with a length of 2. The range of parameters can then be approximated using this bounding box. To clarify, an example has been shown in Figure 6.



lb:	-1	-1	0	$tan^{-1}(2)$	2
sol:	x_o	y_0	p_1	p_2	p_3
ub:	1	1	2	$\pi/2$	≈2.5

lb: lower bound; *ub:* upper bound; *sol*: solution

Figure 6: An example of defining the range of parameters

As the last step, three degrees of freedom for considering the rotation and reflection of the profiles are added to the solution, as shown in Figure 7. These modes create eight regions that are helpful in the parametric modeling of asymmetric profiles. The range of these variables is defined between [-1, 1] so that for values more than 0, the transformations are applied to the profile, and for values lower than 0, no transformation is exerted. The defined optimization problem in this section can be solved by real metaheuristic algorithms. In this paper, FFA is used as it showed more promising performance, especially after adding transformations to the solution.

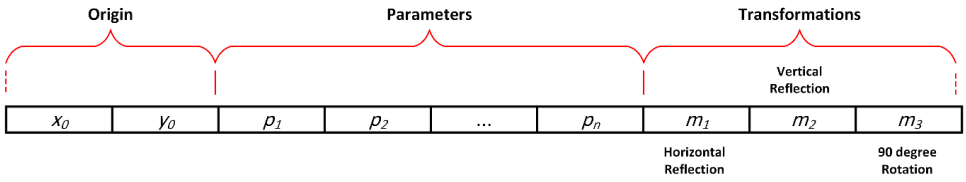


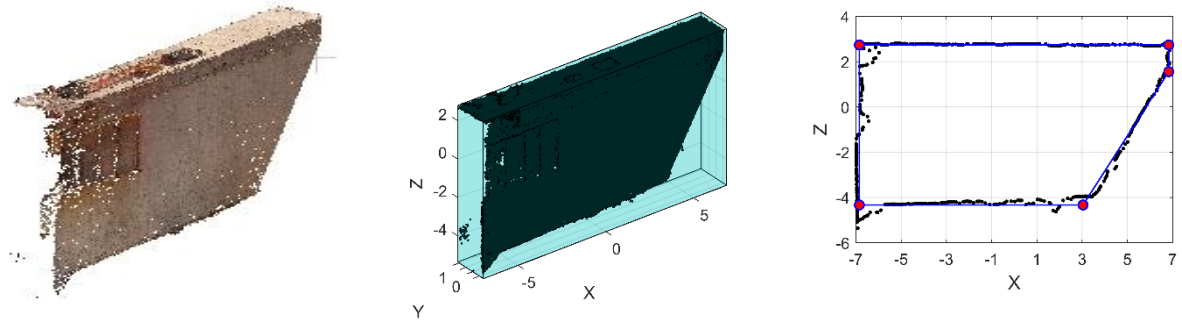
Figure 7: General form of a solution in a metaheuristic algorithm

4. Real-world applications

Two cases are studied to evaluate the performance of the developed methodology on the point cloud of structural elements. The first case is the concrete abutment of a bridge, and the second case is an overpass with two connected wing walls. To validate and compare results, the models are also created manually. The minimum distance of points to the 3-D objects, obtained from our approach and manual modeling, is calculated and finally, a value of *RMSE* is reported in each case.

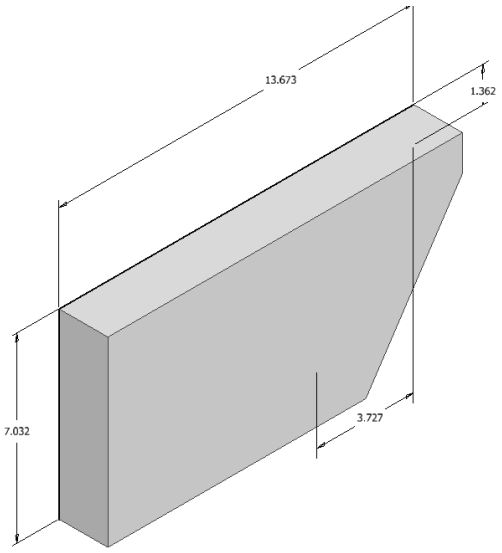
4.1 Case study 1: Abutment

In this case, an asymmetric point cloud of an abutment has been studied. The point cloud included 56,767 points after down-sampling. Due to occlusion, 2 faces out of 7 faces of the element were not present (the bottom and left face). Also, occlusion and clutter could be seen on the remaining faces, especially the back face of the element. To determine the MAABB (see Section 3), PSO was used. It was seen that considering a swarm with 35 particles and 100 iterations is sufficient for solving the problem. c_1, c_2 coefficients were also set 2, and a damping factor of 0.99 was applied. To calculate the area covered by points, an alpha complex with a critical value of alpha for meshing a single region was employed. The area ratios of r_x, r_y, and r_z were computed 0.8833, 0.7531, and 0.8358, respectively. Hence, the y-direction was detected correctly as the extrusion direction. The cross-section of the point cloud was obtained from the alpha hull with the same value of alpha (0.3839). To extract the parameters of the cross-section, FFA with a parametric model of an abutment profile was used. All the points were normalized, and optimization was conducted in this space. The number of 15 fireflies with 30 iterations was initialized. FFA coefficients including β_0, γ, and α were considered 2, 1, and 0.2, respectively. Figure 8 shows the steps of parametric modeling and Figure 9 demonstrates the final output. Comparing the results of our approach and manual modeling illustrates that the proposed approach not only reduce the modelilng time significantly but also might improve the quality of modeling, i.e. lower value of *RMSE*. This can be due to visual errors and rounding numbers that happens in the manual modeling process.



(a) (b) (c)

Figure 8: Parametric modeling process: (a) input PCD; (b) MAABB; (c) Optimized profile

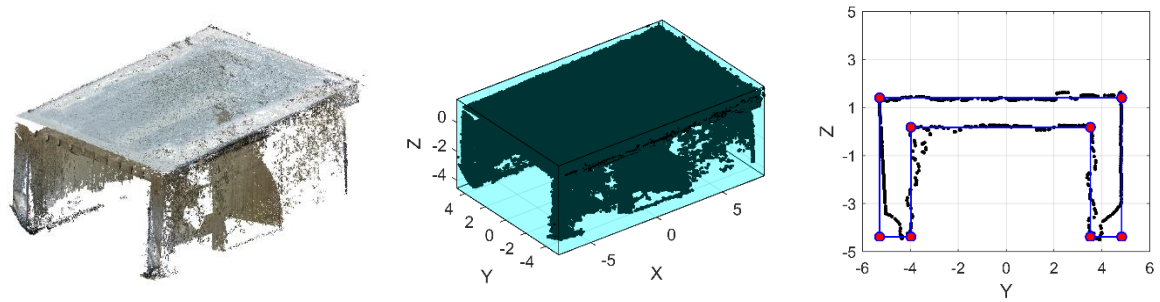


Results	Our approach	Manual modeling
p_1	7.03 m	7.00 m
p_2	13.67 m	13.50 m
p_3	1.36 m	1.40 m
p_4	3.73 m	3.50 m
p_5	1.89 m	1.90 m
RMSE	**0.11 m**	0.25 m
Time	**23.43 sec**	≈ 1200.00 sec

Figure 9: Resulting model of the abutment and its parameters

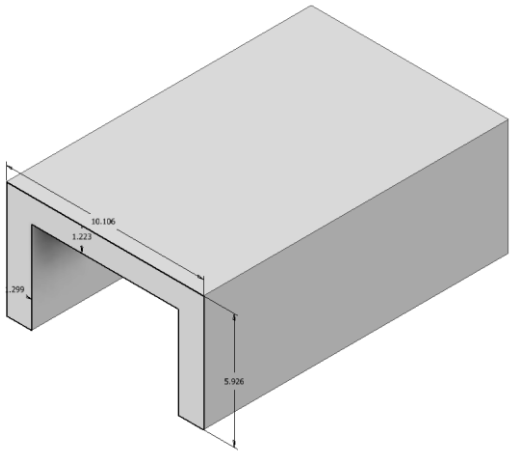
4.2 Case study 2: Overpass with wing walls

In this case, the point cloud of wing walls connected by an overpass has been studied. In contrast to the previous case, this point cloud has an axis of symmetry. Hence, this feature should also be considered in parametric modeling. The point cloud after down-sampling contained 129,028 points. Two faces of each wing wall were not present, and the other faces had clutter and occlusion. All the parameters of the metaheuristic algorithms were considered the same as the first case study. The area ratios of r_x, r_y, and r_z were computed 0.3543, 0.8015, and 0.9569, respectively. Hence, the *x*-direction was recognized as the direction vector. The total time necessary for modeling this structure was 43.67 sec. Figure 10 demonstrates all the steps for automatic parametric modeling based on PCD. Figure 11 also shows the final model and a comparison between the proposed method and manual modeling. As can be seen, the obtained parameters in both cases are very close to each other. However, the accuracy of the model derived from our approach is higher, i.e., lower *RMSE*.



(a) (b) (c)

Figure 10: Parametric modeling process: (a) input PCD; (b) MAABB; (c) Optimized profile



Results	Our approach	Manual modeling
p_1	10.11 m	10.00 m
p_2	5.93 m	6.00 m
p_3	1.22 m	1.30 m
p_4	1.30 m	1.30 m
p_5	15.63 m	15.50 m
RMSE	**0.17 m**	0.38 m
Time	**43.67 sec**	≈ 1200.00 sec

Figure 11: Resulting model of the overpass with wing walls and its parameters

5. Conclusion

In this paper, a method is presented that enables fitting a parametrized bridge model into a given point cloud resulting from a capturing campaign. To this end, metaheuristic algorithms are applied to derive the value of parameters from point cloud data of structural elements. It is shown that these algorithms could extend the conventional *model-based* approach from primitives to more general shapes that are common in infrastructure assets. The presented method consists of three steps: (1) identifying orientation, (2) fitting parametrized cross-section, and (3) applying extrusion operation. In all steps, meta-heuristic optimization approaches were successfully applied. Except for the optimization algorithms' parameters that exist in any

problem, no additional parameter or threshold was set. The accuracy of the proposed method was tested on actual point clouds of structural elements that had a significant amount of occlusion and clutter. The results of the paper show that metaheuristic algorithms can be successfully employed for extracting parameters and deriving the volumetric model of the point cloud. The main advantage of the presented method over existing ones is that a high-quality as-is BIM model is generated with a level of abstraction that fulfills the needs of bridge management systems. In this paper, only components with comparatively simple geometries have been investigated. However, the positive results of the presented feasibility analysis provide grounds for further extending the presented approach to represent the most common bridge types in Germany by highly parameterized models for rapid and automated DT generation from point clouds.

Acknowledgment

The research presented has been performed in the profile of the TwinGen project funded by the German Ministry of Transport and Digital Infrastructure (BMVI).

References

Adán, A., B. Quintana, et al. (2018). "Scan-to-BIM for 'secondary'building components." *Advanced Engineering Informatics* **37**: 119-138.

Bosché, F., M. Ahmed, et al. (2015). "The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components." *Automation in Construction* **49**: 201-213.

Cao, C. and G. Wang (2019). Fitting Cuboids from the Unstructured 3D Point Cloud. *International Conference on Image and Graphics*, Springer DOI: http://dx.doi.org/10.1007/978-3-030-34110-7_16.

Kennedy, J. and R. C. Eberhart (1997). A discrete binary version of the particle swarm algorithm. *1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation*, IEEE.

Kwon, S.-W., F. Bosche, et al. (2004). "Fitting range data to primitives for rapid local 3D modeling using sparse range point clouds." *Automation in construction* **13**(1): 67-81 DOI: http://dx.doi.org/10.1016/j.autcon.2003.08.007.

Laing, R., M. Leon, et al. (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.

Lu, Q., L. Chen, et al. (2020). "Semi-automatic geometric digital twinning for existing buildings based on images and CAD drawings." *Automation in Construction* **115**: 103183.

Lu, R., I. Brilakis, et al. (2019). "Detection of structural components in point clouds of existing RC bridges." *Computer-Aided Civil and Infrastructure Engineering* **34**(3): 191-212 DOI: http://dx.doi.org/10.1111/mice.12407.

Pan, Y., A. Borrmann, et al. (2019). Built Environment Digital Twinning. Report of the International Workshop on Built Environment Digital Twinning presented by TUM Institute for Advanced Study and Siemens AG. Technical University of Munich, Germany.

Qin, G., Y. Zhou, et al. (2021). "Automated Reconstruction of Parametric BIM for Bridge Based on Terrestrial Laser Scanning Data." *Advances in Civil Engineering* **2021** DOI: https://doi.org/10.1155/2021/8899323.

Rocha, G., L. Mateus, et al. (2020). "A scan-to-BIM methodology applied to heritage buildings." *Heritage* **3**(1): 47-67.

Sacks, R., A. Kedar, et al. (2016). SeeBridge information delivery manual (IDM) for next generation bridge inspection, ISARC.

Sacks, R., A. Kedar, et al. (2018). "SeeBridge as next generation bridge inspection: overview, information delivery manual and model view definition." *Automation in Construction* **90**: 134-145 DOI: 10.1016/j.autcon.2018.02.033.

Song, X. and B. Jüttler (2009). "Modeling and 3D object reconstruction by implicitly defined surfaces with sharp features." *Computers & Graphics* **33**(3): 321-330 DOI: http://dx.doi.org/10.1016/j.cag.2009.03.021.
Technion (2015). SeeBridge—Semantic enrichment engine for bridges, Technion**:** 77.
Walsh, S. B., D. J. Borello, et al. (2013). "Data processing of point clouds for object detection for structural engineering applications." *Computer-Aided Civil and Infrastructure Engineering* **28**(7): 495-508.
Yan, Y., B. Guldur, et al. (2017). Automated structural modelling of bridges from laser scanning. *Structures Congress 2017*.
Yang, X.-S. (2008). "Firefly algorithm." *Nature-inspired metaheuristic algorithms* **20**(2008): 79-90.
Zhang, G., P. A. Vela, et al. (2015). "A sparsity-inducing optimization-based algorithm for planar patches extraction from noisy point-cloud data." *Computer-Aided Civil and Infrastructure Engineering* **30**(2): 85-102.
Zhu, Z., S. German, et al. (2010). "Detection of large-scale concrete columns for automated bridge inspection." *Automation in construction* **19**(8): 1047-1055 DOI: http://dx.doi.org/10.1016/j.autcon.2010.07.016.
Zhu, Z., S. German, et al. (2011). "Visual retrieval of concrete crack properties for automated post-earthquake structural safety evaluation." *Automation in Construction* **20**(7): 874-883 DOI: http://dx.doi.org/10.1016/j.autcon.2011.03.004.