Hybrid Workflow for Digital Twin Creation: Human-in-the-Loop Segmentation and Geometry-Aware Modeling of Bridge Structures

MOHAMED MANSOUR, HRISTO VASSILEV
and JORG BLANKENBACH

ABSTRACT

Reliable as-is modeling of existing infrastructure is a cornerstone of structural health monitoring and digital twin creation. While data-driven segmentation approaches have advanced automation for common bridge types, they often struggle to generalize to atypical structures. This paper instead proposes an interactive segmentation and modeling technique that aims to reduce the time needed for producing accurate geometric models and segmentation masks by considering both fine local features as well as large-scale geometry. Our method begins with a graph-based oversegmentation of the input point cloud, compressing millions of points into a superpoint graph optimized for real-time interaction. On this graph, we formulate user-guided segmentation as an energy minimization problem, incorporating both local features and region-growing cues to propagate sparse user input. Once segmented, each component undergoes geometry-specific processing: planar elements are modeled via RANSAC-based decomposition and intersection analysis, while irregular surfaces are reconstructed using adaptive triangulation informed by local point density.The resulting surface representations are converted into IFC-compliant building elements, enabling integration with existing BIM systems. We evaluate the method on the Nibelungen Bridge in Worms, Germany, achieving over 99% segmentation accuracy with minimal user input and a final geometric RMSE of 5.8 cm.

INTRODUCTION

Recent research has advanced geometric modeling of bridges from semantically segmented point clouds through varied approaches. [1] leveraged concave hull algorithms and longitudinal extrusion for bridge decks but struggled with complex elements at semantic boundaries. [2] applied 2D parametric models to bridge cross-sections followed by extrusion, though this required significant domain expertise. [3] employed primitive fitting—using lines for railings and planes for abutments—but only addressed simple spatial relationships between planar elements. These methods show notable limitations when handling complex structures like bridge piers with multiple planar elements and foundation connections, underscoring the need for more sophisticated techniques that can manage intricate spatial relationships while maintaining modeling accuracy.

METHODOLOGY

This section describes the systematic approach for generating geometric-semantic as-is models from point cloud data. The proposed methodology employs a sequential pipeline comprising semantic segmentation and geometric modeling to transform unstructured point cloud data into structured Industry Foundation Classes (IFC) models.

Interactive point cloud segmentatioin

The workflow begins by segmenting a point cloud $\mathscr{P} \in \mathbb{R}^3$, produced by a reality capturing technique. Additional attributes include color as captured by the on-board camera, the intensity of the return signal as well well as multiple local geometric descriptors [4] and the *surface normals* are derived from the covariance matrix of the 10 closest neighbors forming a feature vector $\phi_i \in \mathbb{R}^F$ at each point in $\mathscr{P}$. Directly applying any interactive methods is challenging, because real-time segmentation feedback is difficult on large point clouds and because the extracted feature vectors may be noisy due uncertainties in the capturing process. Therefore, we compress our point cloud by partitioning it into a superpoint graph, where each superpoint is represented by a single constant vector ϕ'. The starting point for partitioning is a k-nearest neighbor graph on the point cloud with edges $\mathscr{E}$. We then employ the cut-pursuit algorithm [5] with a l_0 loss, that is formulated as follows:

$$E(\Phi') = \underbrace{\sum_{i \in \mathscr{P}} ||\phi_i - \phi'_i||^2}_{\text{data fidelity}} + \underbrace{\sum_{(u,v) \in \mathscr{E}} w_{uv} 1\{\phi'_u \neq \phi'_v\}}_{l_0 \text{ regularizer}} \tag{1}$$

In Eq. 1 the quadratic loss between the constant vector ϕ' in each superpoint and the original features is minimized. To counteract the splitting a regularizer penalizes each edge cut by a constant value w_{uv}, which we empirically set to 0.8 in order to produce spatially compact superpoints with an average diameter of $\approx$ 5cm. Note that this behiour is achieved by also including the point coordinates $[x,y,z]_i$ in ϕ_i. By concatenating three-dimensional coordinates (with a larger weight) to each feature vector, the algorithm avoids grouping distant points—even if their colors or normals match.

After the initial partitioning into superpoints we formulate an interactive segmentation approach on the resulting superpoint graph as a maxflow-min-cut problem as follows: For each node (i.e. superpoint) V_i we take the outputs of a small neural net $\theta(\phi'_i) = [D_i(0), D_i(1)] \in \mathbb{R}^2$. This network consists of a single hidden layer with a dropout probability of 0.2 and is trained iteratively as the user is providing positive (1) and negative (0) picks in a hypothetical user interface (see Figure 1). In addition, we define pair-wise potentials between any two neighboring superpoints V_u and V_v in the graph, which we model as an exponential function of their feature difference $\psi_{uv} = \lambda \exp(-\beta ||\phi'_u - \phi'_v||^2)$ thus only penalizing cuts that happen between neighbors with similar features. Finally, we solve for the optimal two-way partition $\{\mathscr{S}, \mathscr{T}\}$ that minimizes the resulting energy function in Eq. 2 via the Boykov-Kolmogorov maxflow/min-cut algorithm [6].

$$E(\mathscr{S}, \mathscr{T}) = \sum_{i \in T} D_i(0) + \sum_{i \in S} D_i(1) + \sum_{\substack{i \in S,\, j \in T \\ (i,j) \in \mathscr{E}}} \psi_{ij} \tag{2}$$

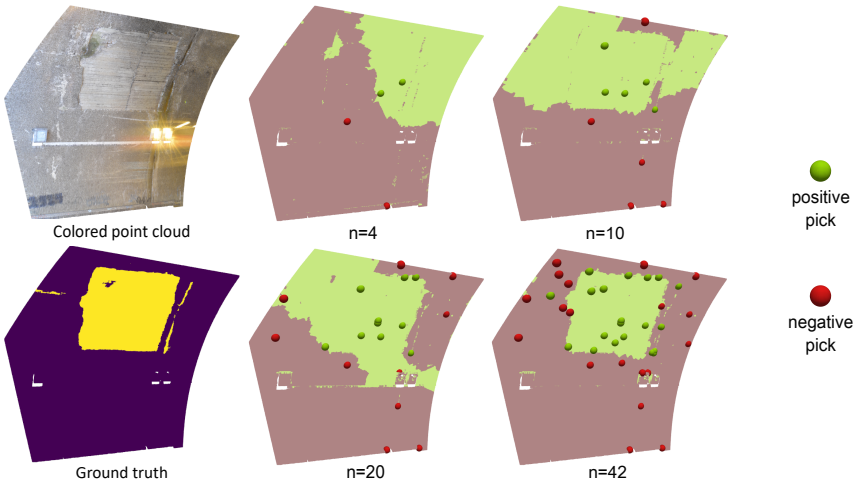

Figure 1. Picking process example on a damaged bridge surface

In Eq.2 we calibrate the smoothing parameters by setting them to $m = \text{median}_{(i,j)\in\mathscr{E}}||\phi'_i - \phi'_j||^2$, $\beta = \frac{\ln 2}{m}$ and $\delta_i = |D_i(0) - D_i(1)|$, $\lambda = \text{median}_i \delta_i$. Our first case study (see Figure 1) shows that this can already provide results of above 95% accuracy after 40 clicks (20 per class) so long as the distinction can be made based on local features. Segmenting an entire bridge column (Nibelungen Bridge, Worms) proved difficult using local features alone. We therefore apply region growing to $\mathscr{P}$, adding unassigned points within $\varepsilon \approx 5,\text{cm}$ of a labeled region if they 1) lie within 3ε of the region's least-squares plane, and 2) have normals within $\alpha = 27°$. Parameters ε and α are data-dependent and estimated via the heuristic of Poux et al. [7]. The region growing results in a segmentation of $r \in [1,\ldots,R]$ regions (Figure 2, left). We can integrate this information into the picking process by defining a lookup table $\theta_{\text{LUT}}(\cdot) : \{0;1\}^R \to \mathbb{R}^{32}$, which maps the region-id to a common feature space that we combine with our geometric features. Our updated formulation for the unary potentials then becomes $[D_i(0), D_i(1)] = \theta_{\text{unary}}(\theta_{\text{local}}(\phi'_i) + \theta_{\text{LUP}}(r_i))$. Using this joint feature set we can see that the mask quickly converges to the correct geometry at 18 total picks and eventually reaches accuracy above 99% (Figure 3). Notably, including both local features and the region growing lookup converges faster than just using the lookup as it helps distinguish edge artefacts found in the region growing process (Figure 2, right).

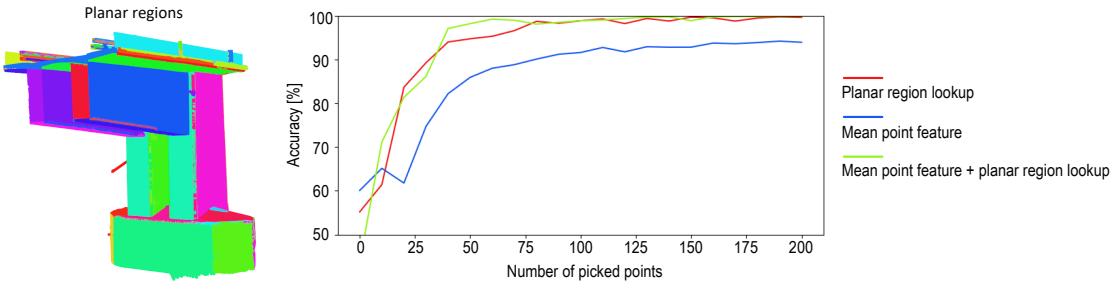

Figure 2. Results of the region growing method *(left)* and the accuracy of a simulated picking process, where points are randomly picked from the wrong classifications. The results are averaged over 30 runs. *(right)*

Support Surface Decomposition

Using the above formulation we extract the points belonging to the ‘support structures‘ and ‘foundation elements‘ classes. The support class, typically comprising multiple planar surfaces, undergoes planar decomposition to identify its constituent surfaces. We employ the Random Sample Consensus (RANSAC) algorithm [8] to iteratively identify and extract planar regions within the foundation point cloud. The RANSAC parameters are (ε = 0.07, min_points = 3, iterations = 1000). The identified planes are then ranked based on their inlier counts and the top 7 planes are retrieved based on specific threshold value, enabling the filtering of outliers and the isolation of surfaces of interest, Figure 4. This hierarchical approach ensures that only significant planar components which constitute the support are retained for subsequent modeling. Each planar surface S_i is defined by the equation $a_i x + b_i y + c_i z + d_i = 0$, where the normal vector $n_i = (a_i, b_i, c_i)$ and d_i is the signed distance from the origin. Planes are grouped by normal orientation into parallel sets (e.g., green and red in Figure 5) to facilitate identification of perpendicular intersections. For group 1 surfaces, we compute the intersection line L of two non-parallel planes S_1 and S_2 by taking the cross product of their normals: $v = n_1 \times n_2$. A point on L is obtained by solving the system of plane equations (e.g., setting $z = 0$), yielding (x_0, y_0, z_0). The line is parametrized as $r(t) = (x_0, y_0, z_0) + tv$. To localize the visible segment, we intersect L with the finite boundaries of both planes, retaining only t-values producing valid in-plane points. The extremal points P_1 and P_2 define the segment endpoints, representing critical corner features for subsequent 3D modeling, as shown in Figure 5.

For the second group (red surfaces), we employ a boundary detection algorithm [9] to extract the outer contour of each surface. To suppress noise and refine the boundary, fuzzy logic filtering is applied [10], followed by an angle-based keypoint detection method. Boundary points are first ordered by their angular position relative to the surface centroid. For each point, vectors to its adjacent neighbors are computed, and the internal angle between them is evaluated; points forming angles less than 155° are classified as corner features. This approach extracts and smooths surface boundaries before identifying key geometric points, as illustrated in Figure 6.

Foundation Surface Decomposition

To accurately model the foundation component, we apply a statistical segmentation

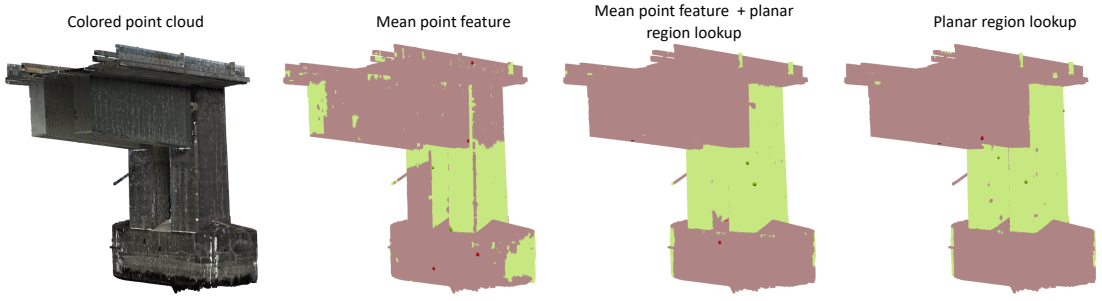

Figure 3. Picking results at $n = 18$ with different input types

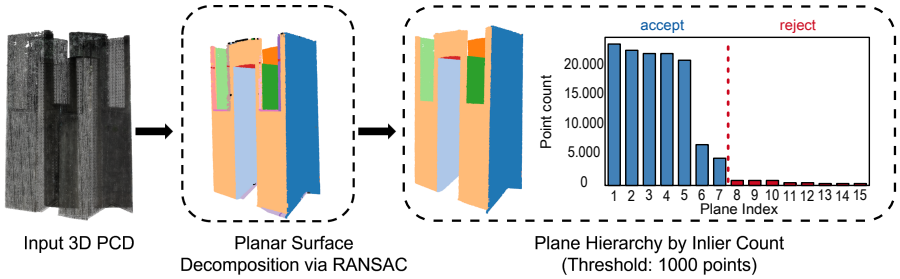

Figure 4. Planar surface decomposition: RANSAC inlier counts by plane index

method using a Gaussian Mixture Model (GMM) [11] fitted to the z-coordinate distribution of foundation points. This allows for the decomposition of the structure into an upper section—representing the interface with support elements—and a lower section in contact with the ground, as shown in Figure 7. For the lower section, we project the points onto the xy-plane and apply the previously described boundary detection and fuzzy logic smoothing techniques to extract the 2D contour, which is used for geometric model generation. Modeling the upper section of the foundation is particularly challenging due to its irregular and unstructured point distribution, which lacks the clear planar features found in the lower section. To reconstruct a coherent surface, we propose an adaptive triangulation method that accounts for local point density and maintains topological consistency. The 3D points are first projected onto the xy-plane while retaining their original z-coordinates, enabling efficient surface reconstruction. Delaunay triangulation [12] is then applied to the 2D projection, producing a high-quality mesh. To refine the triangulation and eliminate spurious connections, we introduce an adaptive edge length filter based on the average nearest-neighbor distance, with a threshold set at five times this mean. Triangles exceeding this threshold are discarded, resulting in a mesh that captures local geometric variations and ensures structural coherence, as shown in Figure 8.

IFC Model Generation

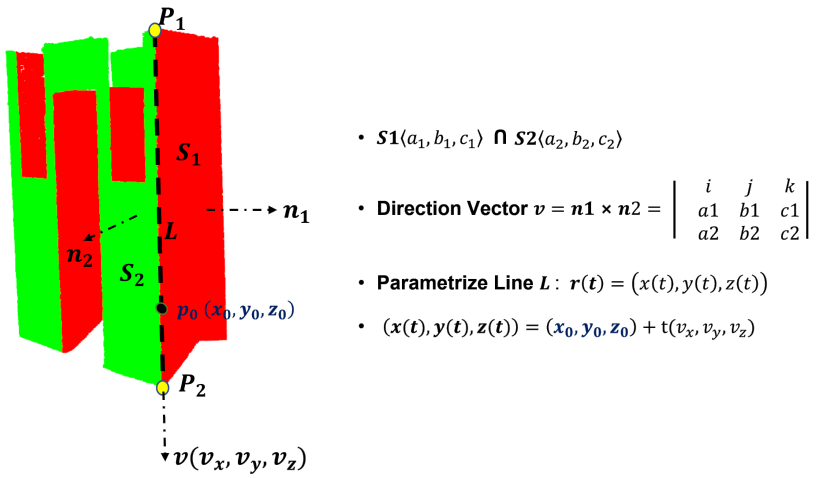

Figure 5. Extraction of intersection line between two non-parallel planar surfaces

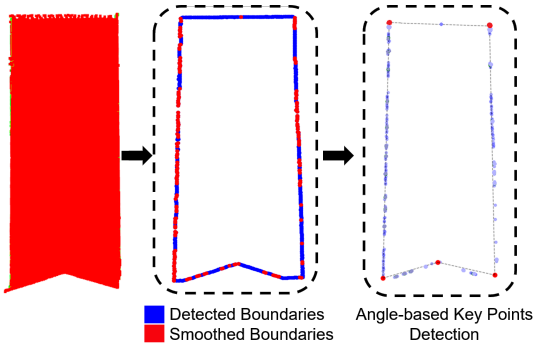

Figure 6. Boundary detection and corner keypoint identification

In the final stage, extracted geometric features are converted into an Industry Foundation Classes (IFC) model using specific strategy for each structural component. For the support element, corner and key points were mapped to `IfcCartesianPoint` and connected via `IfcPolyLoop` to form `IfcFaceOuterBound`, ultimately composing an `IfcFacetedBrep` through an `IfcClosedShell`. The lower foundation section was modeled by defining an `IfcArbitraryClosedProfileDef` from its base boundary points, then extruded vertically using `IfcExtrudedAreaSolid` to generate its volume. For the upper section, the triangulated mesh was preprocessed—by normal orientation, duplicate removal, and smoothing—and then directly converted into an `IfcFacetedBrep`, ensuring a clean and accurate surface representation. The complete IFC model of all components is shown in Figure 9.

RESULTS AND EVALUATION

To evaluate the geometric accuracy of the generated IFC models, we implemented a cloud-to-cloud (C2C) distance-based assessment framework. The models were first converted into 3D mesh representations using the `IfcOpenShell` library. To enable direct comparison with the original point cloud, we applied point subsampling to the mesh surfaces and computed nearest-neighbor distances from each sampled mesh point to the segmented point cloud. These C2C distances quantify geometric deviation, and their distribution was analyzed using several statistical metrics: root mean square error (RMSE) to capture overall deviation with sensitivity to outliers, mean distance for aver-

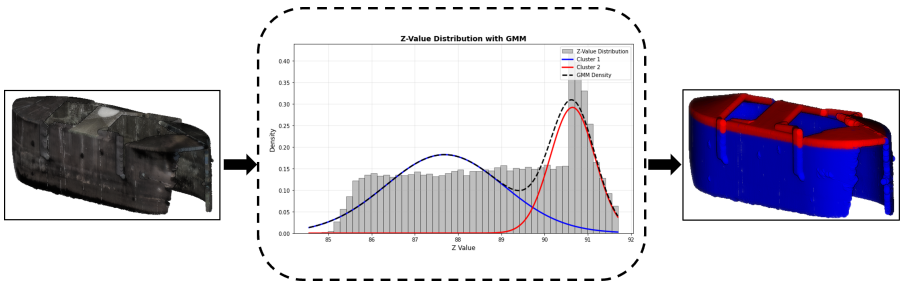

Figure 7. Gaussian Mixture Model segmentation of foundation into upper and lower sections

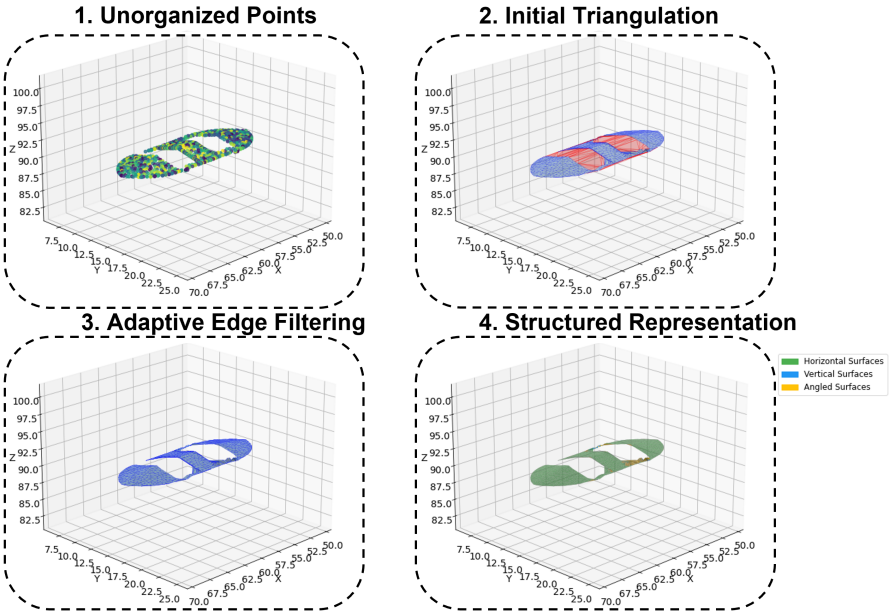

Figure 8. Adaptive triangulation mesh of upper foundation section with edge-length filtering

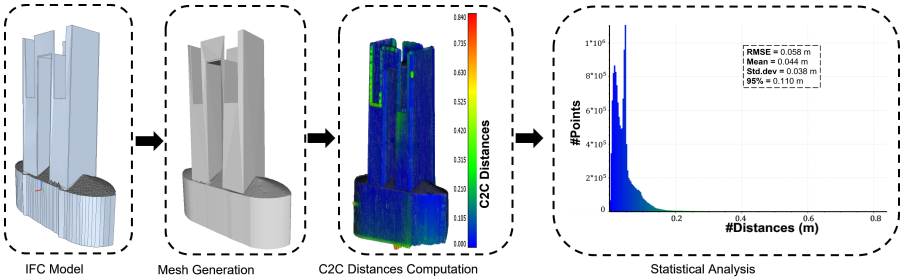

Figure 9. (left) IFC model of the extracted bridge components; (right) cloud-to-cloud evaluation results

age deviation, standard deviation to assess consistency, and the 95th percentile value to characterize the extent of larger deviations. The full evaluation process is illustrated in Figure 9.

CONCLUSION

The workflow evaluation (Figure 9) confirms accuracy with a root mean square error (RMSE) of 0.058 m and 95% of points within 0.110 m of the reference model. While the approach is sensitive to input parameters that require calibration for different architectural contexts, its primary strength lies in accurately representing geometric transitions at boundaries between architectural elements.By combining surface decomposition techniques, boundary detection, corner detection through surface intersections, and adaptive triangulation, the methodology preserves critical architectural features such as openings while establishing meaningful topological relationships. This boundary-aware processing enables faithful modeling of complex interfaces essential for downstream as-

is BIM applications, achieving a balance between geometric fidelity and computational efficiency when processing large-scale architectural point clouds.

ACKNOWLEDGMENT

This project has received funding from the DFG (German Research Foundation) within the framework of "SPP 2388: Hundert plus - Verlängerung der Lebensdauer komplexer Baustrukturen durch intelligente Digitalisierung" with project number 501834640.

REFERENCES

1. Lu, R. and I. Brilakis. 2019. "Digital twinning of existing reinforced concrete bridges from labelled point clusters," *Automation in Construction*, 105:102837.
2. Mafipour, M. S., S. Vilgertshofer, and A. Borrmann. 2023. "Automated geometric digital twinning of bridges from segmented point clouds by parametric prototype models," *Automation in Construction*, 156:105101.
3. Mehranfar, M., H. Arefi, and F. Alidoost. 2021. "A projection-based reconstruction algorithm for 3D modeling of bridge structures from drone-based point cloud," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 46:77–83.
4. Kellner, M., H. Vassilev, A. Busch, R. Blaskow, M. Ferrandon Cervantes, K. N. Poku-Agyemang, A. Schmitt, S. Weisbrich, H.-G. Maas, F. Neitzel, A. Reiterer, and J. Blankenbach. 2024. "Scan2BIM – A Review on the Automated Creation of Semantic-Aware Geometric as-Is Models of Bridges," *avn – allgemeine vermessungs-nachrichten*, (3):159–181, ISSN 0002-5968, doi:10.14627/avn.2024.3.4.
5. Landrieu, L. and G. Obozinski. "Cut Pursuit: Fast Algorithms to Learn Piecewise Constant Functions on General Weighted Graphs," 10(4):1724–1766, ISSN 1936-4954, doi:10.1137/17M1113436.
6. Boykov, Y. and V. Kolmogorov. "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," .
7. Poux, F., C. Mattes, Z. Selman, and L. Kobbelt. "Automatic region-growing system for the segmentation of large point clouds," 138:104250, ISSN 09265805, doi:10.1016/j.autcon.2022.104250.
8. Bolles, R. C. and M. A. Fischler. 1981. "A RANSAC-based approach to model fitting and its application to finding cylinders in range data," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 1981, pp. 637–643.
9. Mineo, C., S. G. Pierce, and R. Summan. 2019. "Novel algorithms for 3D surface point cloud boundary detection and edge reconstruction," *Journal of Computational Design and Engineering*, 6(1):81–91.
10. Warner, J. 2024, "JDWarner/scikit-fuzzy: Scikit-Fuzzy 0.5.0," Zenodo, `https://doi.org/10.5281/zenodo.13372212`.
11. Reynolds, D. A. 2009. "Gaussian mixture models," in *Encyclopedia of Biometrics*, vol. 741, pp. 659–663.
12. Delaunay, B. 1934. "Sur la sphère vide. À la mémoire de Georges Voronoï," *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya*, (6):793–800.