GMP-X Conference on Geometry and Applied Mathematics
April 10–11, 2026 · Marrakesh, Morocco

Projection-Based Structural Element Extraction from Point Clouds

Diana Marin[1] · Julia Reisinger[2] · Peter Kán[1] · Hannes Kaufmann[1]

Abstract

Extracting structural elements from building-scale point clouds is essential for structural assessment, yet remains difficult in industrial environments due to clutter, occlusions, varying density, and missing faces. We propose a fast extraction pipeline that identifies axis-aligned structural elements through orthogonal projections and 2D histograms, and groups peaks into structural candidates via graph-based clustering. The key parameters operate on an explicit histogram representation, making threshold selection visually interpretable for rapid expert iteration. Our method enables rapid conversion of cluttered scans into simulation-oriented structural primitives suitable for downstream structural analysis workflows.

1 Introduction

Industrial buildings represent a large fraction of the existing built environment and are increasingly targeted for adaptive reuse, modernization, and vertical extension. A recurring bottleneck in these workflows is structural assessment: while modern acquisition pipelines, such as photogrammetry, can deliver dense 3D point clouds of entire buildings, structural simulation and verification still rely on analytical models (e.g., beam-and-column skeletons) that are typically created manually. This manual model-building is time-consuming, requires expert knowledge, and is especially error-prone in large facilities where the load-bearing system is embedded in cluttered scenes containing machinery, piping, façades, and partition walls.

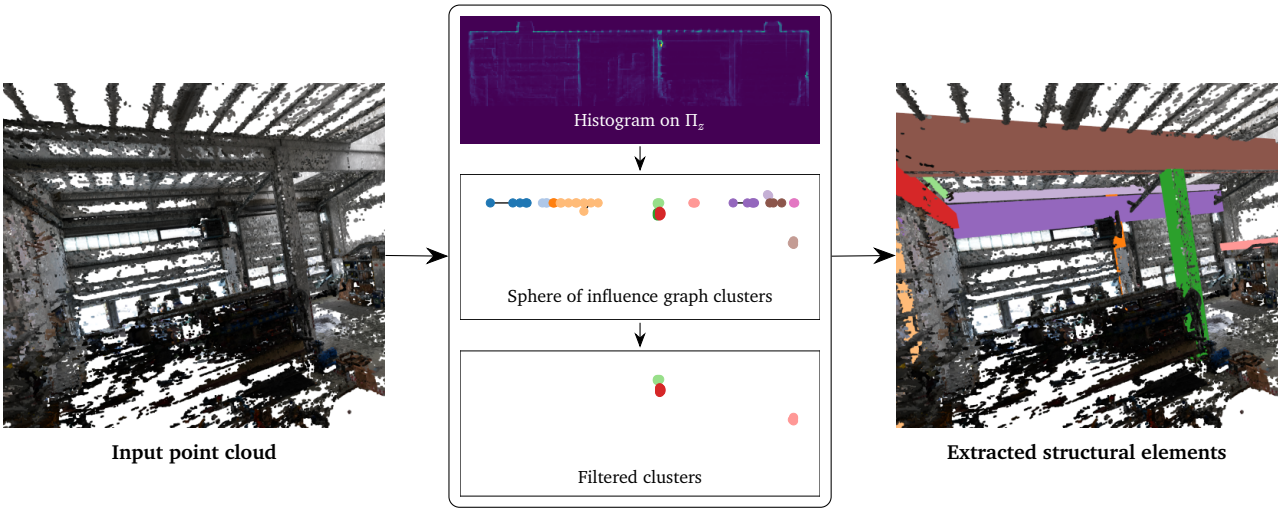


Figure 1: We are able to extract structural elements from noisy point clouds of industrial buildings, through 2D projections and peak clustering in the lower-dimensional space. The histogram values correspond to FEW MANY points in each bin. The dense horizontal lines present at the top of the scanned scene are not structural, and hence, are not detected by our method, showcasing the resilience of our approach.

Most scan-based reconstruction pipelines aim at modelling the entire scene (e.g., Scan to Building Information Model) or at generating volumetric/mesh representations suitable for general finite element meshing. For structural engineering tasks on industrial halls, however, the relevant output is often a selective structural abstraction: a set of columns and beams with correct connectivity and geometric parameters that can be imported directly into structural analysis software. The presence of occlusions, missing faces, and strong non-structural geometry makes this selective extraction substantially more difficult than reconstruction from clean scans of isolated structural components.

We introduce a fast, projection-driven pipeline for automatic extraction of axis-aligned beams and columns directly from full building point clouds - Figure 1. The method leverages the observation that structural elements induce strong axis-aligned density signatures under orthogonal projection, and converts the 3D extraction problem into lightweight 2D projection aggregation

[1]TU Wien Institute of Visual Computing and Human-Centered Technology, Vienna, Austria
[2]TU Wien Institute of Building and Industrial Construction, Vienna, Austria

followed by robust lifting back to 3D. Key parameters act on an explicit and inspectable histogram representation, making threshold choice visually interpretable for rapid expert iteration. Our main contributions are:

1. Projection-histogram peak detection to identify candidate structural elements from dense building scans, replacing expensive accumulator-based line searches with simple and efficient peak extraction.
2. Graph-based clustering of projection peaks (via a Sphere-of-Influence graph [11]) that aggregates peaks into beam/column cross-section candidates, combined with outlier pruning and shape-based filtering to suppress clutter-induced responses.
3. Local geometric filtering in 2D using the neighborhood covariance to extract the corners of wall-encased structural elements.
4. A resulting set of simulation-oriented structural primitives: endpoints, and cross-section parameters (height and width), designed for straightforward export to downstream structural analysis and finite-element workflows.

2 Related Work

Scan-to-BIM and as-built modelling Scan-to-BIM research focuses on reconstructing semantically rich building models from point clouds, typically targeting components such as walls, floors, ceilings, or openings. These pipelines often prioritize completeness and semantic labeling over producing a structurally minimal model suitable for analysis [13, 15, 9]. In industrial buildings, where scans include extensive clutter and occlusions, BIM-oriented approaches frequently require significant preprocessing, manual intervention, or strong priors to separate load-bearing elements from non-structural geometry.

Scan-to-Finite Element Method and point-cloud-to-analysis pipelines Scan-to-FEM pipelines aim to derive geometry suitable for finite element simulation from scan data [14, 8, 12]. A common pattern is to reconstruct dense meshes or volumetric representations and then generate FE meshes for the scanned geometry. The literature also includes slicing- and voxelization-based approaches that work well for relatively clean scenes e.g., heritage structures [10, 1] or tunnels [2] where modelling most visible geometry is the intended result.

Structural element extraction by segmentation and primitive fitting A large class of methods treats structural element extraction as (i) segmentation of points into candidate components (manual or automatic), followed by (ii) primitive fitting (cuboids, cylinders [3], lines) and connectivity inference [6]. In practice, many pipelines become brittle in cluttered scans: segmentation is sensitive to density variation and occlusions, and primitive fitting can be misled by wall fragments, façade elements, and beam-wall intersections. Importantly, several approaches achieve good results only when the structural elements are pre-segmented (often manually) or when scans are restricted to the structural system itself.

Targeted skeleton extraction from full building scans Scan2Beams [7] is the closest method to the present work. It targets the same end goal: automatic extraction of axis-aligned beams and columns from full industrial building scans and export to a structural analysis representation, by discretizing the scene, accumulating projections on axis-aligned planes, detecting strong line carriers, clustering them into elements, and inferring joints by intersections. While effective as a proof of concept, Scan2Beams depends on several user-defined thresholds and a comparatively heavy detection stage (accumulator-based line detection), motivating the present paper's contribution toward faster, histogram-driven detection and streamlined clustering and filtering.

Existing Scan-to-BIM and Scan-to-FEM pipelines either (i) attempt full geometric reconstruction, which is unnecessarily expensive and brittle in cluttered industrial halls, or (ii) rely on strong assumptions such as pre-segmentation or clean scans. Scan2Beams [7] showed that targeted skeleton extraction is feasible directly from hall scans, but its detection stage remains computationally heavy and tuning-sensitive. We close this gap with a projection-driven detector that is substantially faster, and whose key threshold is chosen on an explicit histogram representation, making parameter selection visually interpretable while retaining robustness in in-the-wild scans.

3 Method

We are given a 3D point cloud $P = \{p_i \in \mathbb{R}^3\}$ representing a full building scan that contains both structural and non-structural geometry (e.g., walls, equipment, noise). Our goal is to extract a set of *structural elements* E consisting of beams and columns and to represent each element in a simulation-oriented form (endpoints and cross-section parameters) suitable for downstream structural analysis workflows.

We assume a layout in which the dominant structural system is *axis-aligned*. In practice, this is satisfied in most cases by an additional optional alignment step that estimates a global frame and rotates the point cloud accordingly. After this alignment step, candidate beams and columns are expected to be parallel to one of the principal axes $\{x, y, z\}$.

3.1 Overview

Our method is projection-driven and relies on the observation that axis-aligned structural elements create strong density peaks under orthogonal projection - Figure 2. For each principal axis $a \in \{x, y, z\}$ we:

1. build a 2D projection histogram on the orthogonal plane Π_a and extract the peaks,
2. cluster peaks into possible structure candidates using a proximity graph (Sphere-of-Influence) and reject outliers,
3. apply local geometric filtering to extract corners from wall-like and elongated non-structural elements,
4. lift each cluster back to 3D to fit a parametric beam/column primitive and export its parameters.

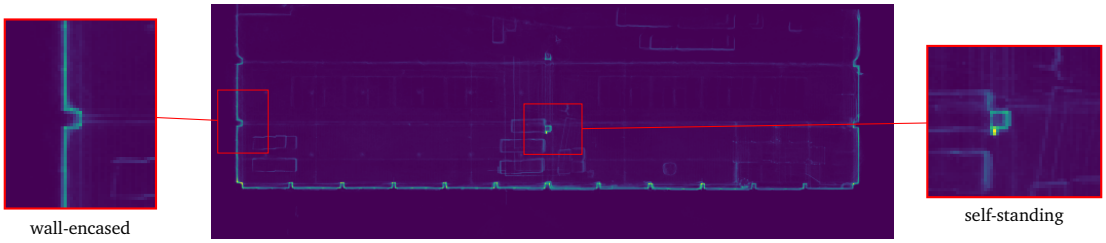


Figure 2: We project an example scene on the XY axes, and showcase examples of structural features highlighted in the histogram as peaks; on the left, structural elements that are part of the wall can be seen, motivating our corner extraction, while on the right, self-standing elements, such as support columns, show up as isotropic structures, motivating our aspect ratio constraints.

3.2 Projection histograms and peak extraction

For each axis $a \in \{x, y, z\}$, we define the orthogonal projection plane Π_a:

$$\Pi_x = yz, \qquad \Pi_y = xz, \qquad \Pi_z = xy,$$

and denote the orthogonal projection of point p by $\text{proj}_{\Pi_a}(p)$.

We discretize Π_a into a regular grid with cell size h and accumulate a 2D histogram

$$H_a(u,v) \;=\; \#\Big\{p \in P \;\Big|\; \text{proj}_{\Pi_a}(p) \in \text{cell}(u,v)\Big\}. \tag{1}$$

Intuitively, high-count bins correspond to locations where many 3D points share the same two coordinates and vary primarily along axis a, which is characteristic of axis-aligned structural elements.

We extract a set of peak locations

$$S_a = \Big\{(u,v) \;:\; H_a(u,v) \geq T \cdot max(H_a)\Big\},$$

where T is a user-defined threshold, between 0 and 1, defining the percentage of the maximum bin count that represents the peak thresholding. Each peak $s \in S_a$ defines a part of an *axis-aligned linear element* in 3D:

$$\ell(s,a) = \Big\{q \in \mathbb{R}^3 \;\Big|\; \text{proj}_{\Pi_a}(q) \in s\Big\}, \tag{2}$$

i.e., a locus of points that share the planar coordinates specified by cell s and are unconstrained along axis a.

3.3 Peak clustering into cross-section elements

A single beam or column typically induces multiple nearby peaks in Π_a, from the various faces and edges that appear in the scan. We therefore cluster peaks S_a, considering each bin's coordinates as vertices, to extract the contour of each element. We build a Sphere-of-Influence (SIG) graph [11] $SIG_a = (V_a, E_a)$ where $V_a = S_a$. Each vertex $s_i \in V_a$ is assigned an influence radius r_i equal to the distance to its nearest neighbor in V_a. Two peaks are connected if their influence regions overlap:

$$(s_i, s_j) \in E_a \quad \Longleftrightarrow \quad \|s_i - s_j\|_2 \leq (r_i + r_j).$$

We chose SIG for clustering because the input density is inherently captured in this graph, requiring no additional parameters, as additionally demonstrated in recent reconstruction works [4, 5]. Connected components of SIG_a yield peak clusters $\{C_a^k\}$. We refine these clusters, based on observations exemplified in Figure 2, using:

- **Outlier pruning:** remove vertices connected only through isolated long edges (i.e., edges much larger than the average edge length in SIG; in our experiments, removing edges larger than $3 \times avg$ yields good results).
- **Shape filtering:** compute the bounding box of each cluster and reject clusters with extreme aspect ratio, which are typical of wall-like or elongated structures. For example, for cluster $\{C_a^k\}$, using the cluster bounding box side lengths (w_k, h_k), reject if $\max(w_k, h_k)/\min(w_k, h_k) > \rho$, with $\rho = 10$ in our experiments. We chose a relatively high aspect ratio because some of the detected clusters contain a structural element surrounded by walls, and we do not want to remove such clusters too early.
- **Corner extraction:** remaining clusters are either containing 3 to 4 sides of a structural element, thus forming a cluster with isotropic contour (similar in shape to a square), or are non-isotropic, containing a corner structure and additional wall elements. We compute the covariance of each point that is part of the non-isotropic clusters, using its 2-ring in SIG. We use the ratio between the small and the large eigenvalues, and keep the largest 15% ratios as the corner points. We remove the rest, hence, removing points where the neighborhood does not exhibit a large variation on both principal axes. We only keep points that are part of corners, removing the additional wall pieces.

The remaining clusters localize candidate beam and column cross-sections on Π_a.

Scene	# Points	Histogram [ms]	SIG [ms]	SIG Filtering [ms]	Corner Filtering [ms]	Total [ms]
Scene 1	5,176,639	546.2	256.4	3.0	0.0	805.6
Scene 2	11,410,836	1616.3	404.1	5.0	7.0	2032.4
Scene 3	22,111,128	2360.7	533.5	2.0	9.0	2905.2
Scene 4	11,410,836	1377.6	317.6	3.4	4.0	1702.6
Scene 5	75,625,894	11827.6	964.0	3.0	0.0	12794.7
Scene 6	54,077,044	5526.7	666.5	1.0	6.0	6200.2
Scene 7	36,500,145	4122.1	472.1	2.0	1.0	4597.2
Scene 8	24,634,975	3089.3	1357.8	25.1	51.5	4523.7

Table 1: Timings of various pipeline parts in milliseconds. The corner filtering shows up as zero in some cases because this step does not get triggered unless there are non-square clusters left to process. We also observe that the histogram processing takes most of the time, which is to be expected, since this is the only step that deals with the full 3D dataset.

Method	1	2	3	4	5	6	7	8
Scan2Beams	39.06	86.53	476.37	129.44	1549.80	585.68	812.82	474.50
Ours	0.81	2.03	2.90	1.70	12.79	6.20	4.60	4.52
Speedup	≈ 48	≈ 43	≈ 164	≈ 76	≈121	≈ 94	≈177	≈ 105

Table 2: Runtime comparison (seconds) between our proposed method and Scan2Beams across all of our datasets. The Scan2Beams runtimes are not directly correlated with the number of points, and rather with the collection of parameters that need to be tuned, since those parameters affect the size of the data structure used internally. The Scan2Beams parameters are chosen to yield the best possible visual results.

3.4 Primitive estimation and output representation

Each cluster C_a^k corresponds to an element aligned with axis a. We estimate:

- **Endpoints:** we find all points that are projected into the cells of C_a^k and around $h/2$ padding around the contour, to mitigate any loss of information due to discretization. We find the minimal and maximal values along axis a to define the endpoints of the current structural element.
- **Cross-section parameters:** Use the projected points of this element in the orthogonal plane Π_a and fit a robust bounding rectangle, yielding width and height parameters.

The final output is a set of structural primitives, $E = \{(e_0, e_1, \text{cross-section})\}$, where (e_0, e_1) are endpoints, and `cross-section` stores the estimated profile parameters. This representation is designed for later usage in downstream structural analysis and finite element workflows (i.e., as node and line-element definitions with associated section properties).

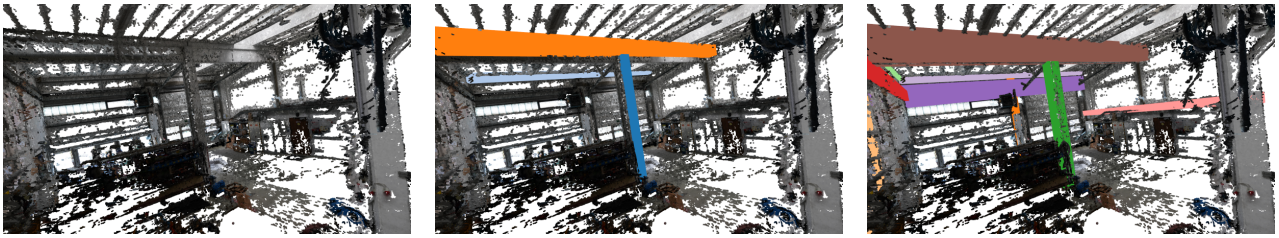
4 Results

We evaluate our method on a collection of 8 in-the-wild point clouds of industrial building interiors captured through photogrammetry. The scans contain substantial non-structural geometry (e.g., walls, machinery, pipes) and exhibit typical artifacts such as occlusions, varying point density, and missing faces. Our scenes contain between $5M$ *and* $75M$ points. The absence of ground truth placements of structural elements prevents us from reporting exact precision/recall results for our datasets. Instead, we emphasize runtime, qualitative results, and focus on our application as a fundamental aid in the process of manual segmentation of such scenes.

Implementation details. All experiments were implemented in Python and run on a device with Intel i9-14900HX and NVIDIA 4090RTX. We report runtimes for the full pipeline and for the main processing stages in Table 1. We use a histogram cell size $h = 0.1$, peak threshold $T = 0.45 - 0.65$, depending on the number of structural elements expected in a given scan.

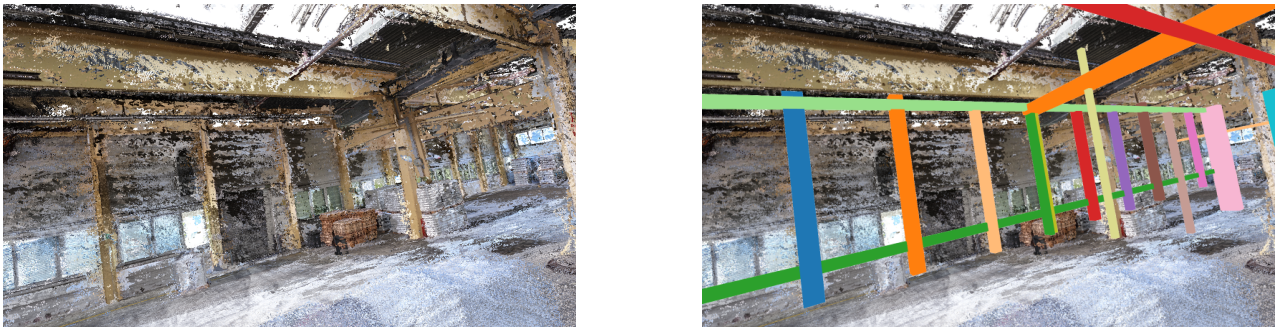
Timings. We compare against Scan2Beams [7], as the prior state-of-the-art in our setting, using their original implementation and parameterization, where possible. For fairness, both methods are applied to the same pre-aligned point clouds and run until the column/beam detection is performed, as the exporting is not time-consuming and can be adapted to various different formats. We report the results in Table 2, where we can observe a speedup of between 43× and 177× over the previous pipeline. We argue that, even if our proposed pipeline still requires 2 parameters to be tuned for good results, the quick results allow for much faster experiments compared to Scan2Beams. Additionally, both parameters can be easily related to the scene (size, density, and number of structural elements), while Scan2Beams requires a collection of absolute parameters that require multiple experiments and cannot easily be connected to the input.

Visual quality and typical outcomes. Figure 4 shows representative results on Scene 5. Our method reliably extracts prominent load-bearing columns and main beams, even in the presence of clutter and partial visibility. The vertical structures are correctly extracted, except for the pillar on the right, which was not captured densely enough, and thus, there is not enough data to



(a) Original capture. **(b)** Scan2Beams [7]. **(c)** Ours.

Figure 3: Qualitative comparison between our method and Scan2Beams on Scene 2.



(a) Original capture. **(b)** Our results, as an overlay for better visibility.

Figure 4: Example of extracted elements in Scene 5.

discover this column as a peak in the histogram. In such cases, a local density threshold T_local, which is currently left as future work, could help detection in very sparsely captured areas. Moreover, the horizontal beam close to the roof is only partially detected by our algorithm, due to the sparsity of the capture close to the roof. In Figure 3, we present extraction results on Scene 2, where Scan2Beams fails to detect some of the horizontal beams.

Failure modes. We observe three failure cases: (i) *missed* elements under severe occlusion or sparse sampling (potentially alleviated by local or adaptive density thresholds), (ii) *false positives* when dense clutter aligned with principal axes (e.g., well-captured pipes) survives filtering; this could be reduced by length and/or size limits and simple structural priors on plausible placement (e.g., enforcing horizontal beams to only be placed on top or close to the top of the vertical ones), (iii) *violated assumptions* in non-axis-aligned structures (Figure 5); such scans could be partitioned into approximately axis-aligned subsets and processed separately.

4.1 Discussion

Our proposed method prioritizes scalability and fast structural element extraction. In practice, we find that the extracted element set is suitable as a *first-pass structural skeleton*, after which limited manual refinement can yield an analysis-ready model.

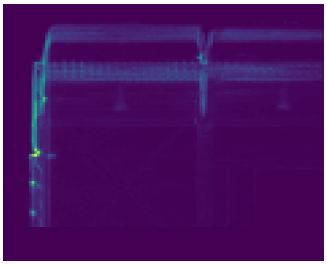
Impact of parameters. Our pipeline exposes only two dataset-dependent parameters: the histogram resolution h and threshold T (Section 3), while the others are fixed and relate to the method. The dataset-dependent parameters govern discretization and peak selection. Smaller h preserves detail but can fragment peaks under noise or occlusion, whereas larger h stabilizes accumulation at the risk of merging nearby elements. Given the histogram, T can be selected in an interpretable space: it directly controls which prominent structures in the 2D accumulation are retained, allowing experts to include or exclude marginal traces with immediate feedback, creating an explainable extraction pipeline. In contrast, Scan2Beams relies on several thresholds applied to intermediate stages whose scale depends on point density, grid resolution, and clustering, so discretization changes can shift the meaning of parameters. By anchoring decisions to an explicit, inspectable histogram, our method reduces tuning effort and enables rapid expert iteration (Table 1), particularly valuable for large in-the-wild scans where turnaround time is critical.

4.2 Limitations and future work

Despite the speed improvements, our method does not yet recover all beams and columns in highly cluttered scans. Severe occlusions, missing faces, or ambiguous projections can lead to missed elements, so limited manual correction may still be required for engineering-grade completeness; nevertheless, the overall effort remains far below fully manual modeling and prior pipelines.

A second limitation is that cross-section estimation is currently simplified rather than being a dedicated, robust reconstruction step. A natural next step is to add a cross-section fitting stage (e.g., parametric profile fitting on filtered inliers in the orthogonal plane) after candidate extraction, improving geometric accuracy and exported simulation parameters.

(a) Example of broken assumptions.



(b) Histogram on Π_y projection.

Figure 5: Scene 7 reconstruction, where our axis-aligned assumptions fail partially on one axis projection, due to the roof being formed of two parallel sections.

Future work further includes relaxing the strict axis-alignment assumption (slanted members and local deviations), reducing parameter sensitivity via adaptive thresholding, and adding automatic joint classification to support direct import into structural engineering toolchains.

5 Conclusion

We presented a fast, projection-driven pipeline for extracting axis-aligned beams and columns directly from full building point clouds. The method leverages orthogonal projection histograms to obtain strong structural elements efficiently, clusters peak responses into location hypotheses using a graph-based grouping strategy, and refines candidates via local covariance descriptors to suppress wall-like and clutter-induced structures. The resulting elements are produced in a simulation-oriented representation (axis, endpoints, and associated parameters), enabling streamlined integration with structural analysis workflows. In practical building-scale scans, the approach substantially reduces the time-to-model compared to previous works while maintaining robustness to clutter and partial visibility.

Acknowledgments

This work was supported by the Austrian funding institution FFG as part of the ‘RE:STOCK INDUSTRY’ project. We would like to additionally thank our project partners for providing the use cases.

References

[1] V. S. Alfio, D. Costantino, M. Pepe, and A. Restuccia Garofalo. A Geomatics Approach in Scan to FEM Process Applied to Cultural Heritage Structure: The Case Study of the "Colossus of Barletta". *Remote Sensing*, 14(3), 2022.

[2] L. Cui, L. Zhou, Q. Xie, J. Liu, B. Han, T. Zhang, and H. Luo. Direct generation of finite element mesh using 3D laser point cloud. *Structures*, 47:1579–1594, 2023.

[3] Y.-J. Liu, J.-B. Zhang, J.-C. Hou, J.-C. Ren, and W.-Q. Tang. Cylinder detection in large-scale point cloud of pipeline plant. *IEEE transactions on visualization and computer graphics*, 19(10):1700–1707, 2013.

[4] D. Marin, S. Ohrhallinger, and M. Wimmer. SIGDT: 2D Curve Reconstruction. *CGF*, 41(7):25–36, 2022.

[5] D. Marin, S. Ohrhallinger, and M. Wimmer. Parameter-Free Connectivity for Point Clouds. In *VISIGRAPP (1): GRAPP, HUCAPP, IVAPP*, pages 92–102, 2024.

[6] F. Noichl, A. Borrmann, and Y. Pan. Automated Steel Structure Model Reconstruction through Point Cloud Instance Segmentation and Parametric Shape Fitting. *Journal of Information Technology in Construction*, 30:1099–1122, 07 2025.

[7] J. Reisinger, D. Marin, P. Rufinatscha, and P. Kan. Scan2Beams: Moving Towards Automated Modelling and Analysis of Structural Industrial Building Stock. In *Proceedings of the 2025 European Conference on Computing in Construction*, volume 6 of *Computing in Construction*, Porto, Portugal, July 2025. European Council on Computing in Construction.

[8] B. Riveiro, G. Cubreiro, B. Conde, M. Cabaleiro, R. Lindenbergh, M. Soilán, and J. Caamaño. Automated Calibration of FEM Models using LIDAR Point Clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:969–974, 2018.

[9] R. Rolin, E. Antaluca, J.-L. Batoz, F. Lamarque, and M. Lejeune. From Point Cloud Data to Structural Analysis Through a Geometrical hBIM-Oriented Model. *J. Comput. Cult. Herit.*, 12(2), May 2019.

[10] Z. Selman, J. Musto, and L. Kobbelt. Scan2FEM: from Point Clouds to Structured 3D Models Suitable for Simulation. In *GCH*, pages 1–10, 2022.

[11] G. T. Toussaint. A graph-theoretical primal sketch. In *Machine Intelligence and Pattern Recognition*, volume 6, pages 229–260. Elsevier, 1988.

[12] W. Xu and I. Neumann. Finite Element Analysis based on A Parametric Model by Approximating Point Clouds. *Remote Sensing*, 12(3), 2020.

[13] L. Yang, J. C. Cheng, and Q. Wang. Semi-automated generation of parametric BIM for steel structures based on terrestrial laser scanning data. *Automation in Construction*, 112:103037, 2020.

[14] K. Yu, C. Zhang, M. Shooshtarian, W. Zhao, and J. Shu. Automated finite element modeling and analysis of cracked reinforced concrete beams from three dimensional point cloud. *Structural Concrete*, 22(6):3213–3227, 2021.

[15] S. Zbirovský and V. Nežerka. Open-source automatic pipeline for efficient conversion of large-scale point clouds to IFC format. *Automation in Construction*, 177:106303, 2025.