www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

AUTOMATED STEEL STRUCTURE MODEL RECONSTRUCTION THROUGH POINT CLOUD INSTANCE SEGMENTATION AND PARAMETRIC SHAPE FITTING

SUBMITTED: March 2025
REVISED: June 2025
PUBLISHED: July 2025
EDITOR: Robert Amor
DOI: 10.36680/j.itcon.2025.045

Florian Noichl, M.Sc.
Chair of Computing in Civil and Building Engineering, TUM Georg Nemetschek Institute Technical University of Munich, Germany
ORCID: https://orcid.org/0000-0001-6553-9806
florian.noichl@tum.de

Yuandong Pan, Dr.-Ing.
Department of Engineering, University of Cambridge, United Kingdom
ORCID: https://orcid.org/0000-0002-5331-6901
yp296@cam.ac.uk

André Borrmann, Prof. Dr.-Ing.
Chair of Computing in Civil and Building Engineering, TUM Georg Nemetschek Institute, Technical University of Munich, Germany
ORCID: https://orcid.org/0000-0003-2088-7254
andre.borrmann@tum.de

SUMMARY*: This paper presents an automated method for converting laser-scanned point cloud data of steel structures into accurate digital 3D models. Point cloud data from laser scanning in such environments typically contains gaps, occlusions, and noise that complicate precise digital reconstruction of complex steel frameworks. Our approach addresses these limitations by combining geometric feature analysis with skeleton-based topology preservation. The method identifies individual steel beam instances within the point cloud, determines precise beam orientations through iterative model-based algorithms, and reconstructs occluded sections using skeletal representations. Cross-sectional profiles are matched to standardized catalogs through multi-objective optimization, generating complete 3D models in IFC format. Validation on industrial point cloud data of I-beam steel structures demonstrates high accuracy, achieving a mean angular precision of 0.1° for beam orientation and a mean geometric deviation of 4.1 mm between source data and reconstructed models. The method maintains robust performance across varying point densities and partial occlusions. This technology addresses critical needs in the construction and manufacturing industries. Potential applications include automated as-built documentation, construction quality control, retrofit planning for existing structures, and generation of digital models for infrastructure management. The automated processing eliminates manual interpretation bottlenecks in point cloud workflows, reducing processing time significantly.*

KEYWORDS*: Scan-to-BIM, point cloud, material inventory, instance segmentation, multi-objective optimization, procedural geometry, model reconstruction.*

REFERENCE*: Florian Noichl, Yuandong Pan & André Borrmann (2025). Automated Steel Structure Model Reconstruction through Point Cloud Instance Segmentation and Parametric Shape Fitting. Journal of Information Technology in Construction (ITcon), Vol. 30, pg. 1099-1122, DOI: 10.36680/j.itcon.2025.045*

1. INTRODUCTION

Along the entire lifecycle of facilities in the built environment, stakeholders involved in the AECO industry (Architecture, Engineering, Construction, and Operations) can benefit from digital models, given they are up-to-date and accurate with regard to the required information. With the introduction and widespread application of BIM (Building Information Modeling), this fact has received great attention for the planning and construction of buildings (Borrmann et al., 2018). For existing building stock without digital models, data needs to be collected and transformed into a usable format, which is expensive and time-consuming (Fumarola & Poelman, 2011). Similarly, facilities are, in general, subject to updates and changes, so the respective digital models need to be kept up to date to preserve their full value (Bosché et al., 2015). Research on the topic of Scan-to-BIM aims to solve this issue by automating the process to make the relevant information available for efficient renovation, maintenance, and facility management (Son, Kim, & Turkan, 2015; Q. Wang, Guo, et al., 2019).

The technology involved in the Scan-to-BIM Process has evolved significantly in recent years. Patraucean et al. (2015) provide a comprehensive overview and highlight key challenges for automation in as-built modeling, including clutter, occlusions, and other reasons for data incompleteness. These challenges are particularly evident in industrial settings, where complex geometries, dense environments, and the presence of specialized equipment create a challenging environment.

Industrial facilities present specific challenges for Scan-to-BIM implementation, specifically in Mechanical, Electrical, and Plumbing (MEP) systems. Son, Kim, & Kim (2015) address the intricacies of modeling industrial equipment, emphasizing the need for high precision and the utility of prior knowledge from CAD data. To capture suitable data for industrial facilities, laser scanning is the most suitable way to acquire site conditions with high precision (Fröhlich & Mettenleiter, 2004). Within industrial environments, elongated structural elements with various cross-sections – such as beams and columns – present the greatest difficulty for manual modeling tasks, necessitating careful consideration of their specific features and challenges (Agapaki et al., 2018). Current approaches to this problem have significant limitations: some restrict model generation to center lines only (Smith & Sarlo, 2021), some handle only simple cylindrical objects (Yang et al., 2020), while others simplify complex cross-sections into cuboids using oriented bounding boxes (OBBs) (Justo et al., 2023). Other methods require consistent occlusion patterns along the length of elongated objects for their segmentation algorithms (Yan & Hajjar, 2022) or for their orientation estimation techniques to function properly (Yang et al., 2020).

Automating steel structure reconstruction from point clouds remains unsolved, particularly when dealing with complex geometries that cause heavy occlusions in captured data. Previous approaches have struggled with issues such as unevenly distributed occlusions and have not addressed the identification of standardized cross-sections or export to open exchange formats for BIM models.

Despite advances in point cloud processing and 3D reconstruction, a critical research gap exists in developing comprehensive automated methods can simultaneously: (1) handle semantically segmented point clouds with significant, inhomogeneous occlusions typical of industrial environments, (2) accurately identify and fit standardized steel profile cross-sections to individual beam instances, and (3) generate industry-standard IFC-compliant BIM models that maintain geometric fidelity while ensuring interoperability across different software platforms.

This paper addresses these challenges by introducing a novel process that leverages both local and aggregated geometric features to identify and separate individual beam instances. Our approach combines data-driven and model-based techniques to generate detailed, interoperable, and partially parametric procedural 3D models from point clouds that have been semantically segmented into the investigated component type. It employs multi-objective optimization to ensure optimal fitting of standardized steel profile cross-sections to segmented instances, even with significant occlusions.

The key contributions of this paper are:

1. An automated Scan-to-BIM method generating IFC models for steel structures from semantically homogeneous point cloud data, maintaining high fidelity to the original scan geometry
2. Introduction of two methods to retrieve the orientation of elongated objects at both local and segment levels
3. A multi-objective optimization approach that identifies the best-matching cross-section profile for each

beam instance, balancing geometric accuracy with standardized specifications

4. Experimental validation demonstrating the method's efficacy on real-world industrial point cloud data with high geometric complexity and significant, inhomogeneous occlusions.

The remainder of this paper is organized as follows: Section 2 reviews relevant literature, Section 3 details our proposed methodology, Section 4 presents experimental validation and results, and Section 5 concludes with a discussion of findings and future directions.

2. RELATED WORKS

The domain of Scan-to-BIM has seen various attempts to improve and extend capabilities for broader applications. The solutions developed are diverse and varied; some introduce individual processing steps while others aim for comprehensive, end-to-end methods. There are some basic processing steps that many methods have in common. To structure the following section on related works, the presented methods are grouped into general works on the overarching topic of Scan-to-BIM, specific methods for point cloud processing through Semantic and Instance Segmentation, Skeleton Extraction and Model Reconstruction. Each domain is covered by the most relevant publications to keep this overview brief for such a wide variety of covered topics.

2.1 Scan-to-BIM

The term Scan-to-BIM describes the process of converting 3D point clouds, usually laser scans, of existing buildings or structures into detailed BIM models, allowing architects, engineers, and other AECO professionals to use this data for further activities in planning, management, and others (Son, Kim, & Turkan, 2015; Volk et al., 2014). Recent research has made substantial progress in developing end-to-end Scan-to-BIM solutions.

Developments in Scan-to-BIM are often focused on the reconstruction of structural building models. Ochmann et al. (2016) introduce an automated method for reconstructing volumetric building models (walls and slabs) based on indoor point clouds. Other researchers approach the problem using semi-automatic solutions, combining automatic processes with user interaction for complex modeling tasks (Macher et al., 2017). Mehranfar et al. (2024) reconstruct BIM models with high accuracy, restricted to Manhattan-world structures. Won Ma et al. (2024) present a method that allows one to create models including structural and non-structural objects with high accuracy. All three of the aforementioned methods are automated methods that introduced significant novelty to the field of Scan-to-BIM. Some works shift focus from major structural parts to finer granularity and aim to automatically recognize and model small elements (Adán et al., 2018; Pan, Braun, et al., 2022).

As mentioned above, the main focus of this work is on the domain of industrial buildings, which poses specific challenges. Agapaki et al. (2018) systematically investigate different objects and shapes to prioritize developments in Scan-to-BIM for industrial facilities, later providing a compact categorization of relevant shapes by the cross-section of extruded profiles (CLOI, Agapaki et al., 2019). Yang et al. (2020) present an advanced method to reconstruct a specific type of steel truss with high detail, including connecting parts and curved elements. While the level of detail of this work is unparalleled, it comprises shortcomings in terms of automation, as several processing steps require manual intervention; Furthermore, the high level of detail is limited to a very specific connection type between cylindrical members of the structure, limiting the ability of this semi-automated method to generalize to other datasets (Yang et al., 2020). Geometric reconstruction of elongated parts requires defining center lines for sweep and extrusion operations. Some methods focus on this aspect rather than performing full model reconstruction, returning structural beam lines or skeletons (see Section 2.3) from point clouds based on several assumptions about input data structure and orientation (Smith & Sarlo, 2021; Yan & Hajjar, 2021). However, 3D models of structural lines alone lack information for further processing, such as collision detection and quantity takeoff. To create such methods, cross-section shapes need to be determined for each elongated object. To address this challenge in the presence of significant occlusions, Yan & Hajjar (2022) process a dataset with known scanning locations to apply voxel occlusion labeling by ray tracing. While this is the most advanced published method with regard to cross-section fitting, it fails to retrieve cross-section type for these standardized profiles, and the applied segmentation method is dependent on lengthwise consistent occlusions (Yan & Hajjar, 2022). The work by B. Wang et al. (2021) represents a significant step towards full automation in Scan-to-BIM solutions for industrial buildings, with a precise method to generate parametric models of complex MEP scenes. However, this approach requires an as-designed model for several processing steps and is, therefore, not considered as a pure Scan-to-BIM method but rather connected to the neighboring field of Scan-vs-BIM, which assumes the availability of a BIM model of the building (Abreu et al., 2023; Bosché et al., 2015).

Similar to industrial buildings, many bridges feature components made of extruded profiles. Recently developed works in Scan-to-BIM for infrastructure have made significant strides in addressing the unique challenges posed by these structures, including the methods mentioned above developed by Yan & Hajjar. Justo et al. (2023) apply automated segmentation and classification specifically for bridge components with extruded profiles. Although their highly advanced approach includes the creation of structural graphs and an export to IFC (Industry Foundation Classes, ISO 16739-1:2024), it remains ambiguous about geometric detail as bridge components are merely considered as OBBs (Justo et al., 2023).

To provide a transparent comparison, Table 1 evaluates the most relevant end-to-end Scan-to-BIM methods for elongated parts across five key technical aspects that distinguish advanced methods in this field: automated segmentation, automated 3D model reconstruction, non-primitive cross-section handling, robustness to inconsistent occlusions along objects, and fully open-source availability of implementation dependencies. These aspects were selected because they capture both the automation in critical processing steps and the output modalities that differentiate state-of-the-art approaches from conventional methods (as detailed in Section 1). The comparison includes the novel method presented in this paper alongside the most relevant existing approaches.

Table 1: Comparison of end-to-end Scan-to-BIM methods for elongated parts, based on key technical aspects.

Source	Automated segmentation	Automated 3D model reconstruction	Non-primitive cross-sections	Robustness to inconsistent occlusions along objects	Fully open-source
(Yang et al., 2020)	✗	✗	✓	✓	✗
(Smith & Sarlo, 2021)	✗	✗	✓	✓	✓
(Yan & Hajjar, 2022)	✓	✗	✓	✗	✓
(Justo et al., 2023)	✓	✓	✗	✗	✓
Ours	✓	✓	✓	✓	✓

2.2 Semantic and Instance Segmentation

Semantic segmentation is a crucial step in the Scan-to-BIM process, as it allows the automated classification of point regions in a point cloud into pre-defined semantic categories. Grilli et al. (2017) and Xie et al. (2020) provide a comprehensive review of point cloud segmentation and classification algorithms, including edge-based, region-growing, model-fitting, and machine learning methods. One challenge for semantic segmentation for industrial facilities is that there are no publicly available datasets representing the important class categories for such facilities (Cazorla et al., 2021; Gao et al., 2021). Yin et al. (2021) have addressed this issue by creating a bespoke network architecture for semantic segmentation and training it on their own annotated dataset. Agapaki et al. (2019) have collected and reported a dataset in line with the most important object and cross-section types, as mentioned in Section 2.1. However, this dataset is not publicly available. To address the challenge of insufficient training data, synthetic data integration presents a viable alternative. This approach has been demonstrated by Ma et al. (2020) for conventional indoor environments within the S3DIS dataset (Armeni et al., 2016), and subsequently extended by Noichl et al. (2024) to industrial plant contexts.

Instance segmentation goes beyond semantic segmentation by not only classifying points but also distinguishing individual instances of objects within the same class (Han et al., 2020). This step is crucial for identifying specific building components and thus obtaining the necessary information for model reconstruction, particularly in complex industrial environments with multiple similar, often connected elements. However, the lack of domain-specific training data and the parametric nature of elongated parts prohibit the robust application of Machine Learning methods for instance segmentation in the industrial domain. Justo et al. (2023) solve the problem by applying PCA (Principal Component Analysis) to identify instance orientation and the fitting and further processing OBBs. Campagnolo et al. (2023) apply RANSAC (Schnabel et al., 2007) and DBSCAN (Borah & Bhattacharyya, 2004) to distinguish instances of structural building elements. Similarly, Chen et al. (2021) perform hierarchical clustering to aggregate points to consistent instance clusters. Agapaki & Brilakis (2021) use geometric primitives and machine learning techniques to separate instances of industrial components. Several approaches solve the segmentation problems through projection and subsequent application of pattern matching or similar algorithms in 2D (Smith & Sarlo, 2021; Souza et al., 2019). A successful application of such projection-based methods requires precise estimation of the direction of elongated components, robust to noise and occlusions. Several methods circumvent this challenge by using the assumption of axis-aligned structures and performing projections along the axes of the coordinate system (Souza et al., 2019; B. Wang et al., 2021).

2.3 Skeleton Extraction and Model Reconstruction

To understand and further process 3D geometries, it is helpful to identify center line structures, often denoted as *skeletons*. There are various ways to obtain such skeletons from 3D geometry, represented in the form of mesh or point clouds; Tagliasacchi et al. (2016) provide a comprehensive, generic method overview. For application in Scan-to-BIM, Justo et al. (2023) use the identified OBBs to extract centerlines and further processes to generate a skeleton representation as a structural graph. Q. Wang, Tan, et al. (2019) retrieve a line-based representation of the point cloud in which the lines do not represent the center line but rather the edges that describe the limiting surfaces of the building's interior. Pan, Noichl, et al. (2022) use center lines of RANSAC-detected cylindrical shapes for the model reconstruction of pipe runs, B. Wang et al. (2021) derive the skeleton representation from slices of the point cloud after model-based detection of elements, refined by estimated connection types, similar to the pattern-matching approaches applied by Smith & Sarlo (2021).

Elongated parts like pipes, columns, and beams, which have consistent cross-sections (see Figure 1), can be efficiently modeled using simple procedural modeling techniques. For conventional steel beams with straight axes, a basic extrusion operation suffices. This approach, known in general as a sweep operation, is particularly effective for creating 3D models of objects with uniform cross-sections along a defined path (Borrmann & Berkhahn, 2018). Amongst others, Pan, Noichl, et al. (2022) and B. Wang et al. (2021) perform sweep operations along paths composed of straight and curved sections for straight pipe segments and elbows. Yan & Hajjar (2022) further consider changing cross-sections along the skeleton axis. Extruding rectangular cross-sections leads to cuboid-shaped objects in the reconstructed model. Justo et al. (2023) use refined OBBs for 3D model reconstruction directly. Complex, immutable geometries can be inserted from suitable catalogs after detection in the scene (B. Wang et al., 2021). Other model types comprise complex shapes controlled by parameters that can be obtained through instance enrichment and subsequent optimization to find optimum sets of parameters to fit the input data (Mafipour et al., 2023).

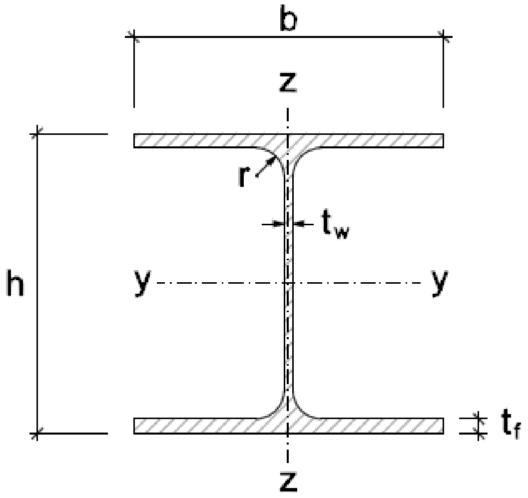


Figure 1: Notation for I-shaped profiles according to European standard EC 3-1-1.

Scan-to-BIM approaches aim to transform captured data into semantically rich, use-case-ready digital models with high degrees of automation. While significant progress has been made across various structure and building element types, a relevant gap remains in automated processing and reconstruction of steel structures with standardized profiles, which are particularly important in industrial environments. Existing methods are limited by their inability to handle inconsistent occlusions along elongated objects, accurately identify standardized cross-section profiles (such as the I-shaped profile depicted in Figure 1), or generate geometrically precise BIM models in open exchange formats. Methods like the one presented by Yang et al. (2020) achieve highly detailed output but lack automation, while the approaches by Yan & Hajjar struggle with inconsistent occlusions, and Justo et al. (2023) oversimplify their geometric reconstruction to OBBs. Despite several works highlighting the importance of extruded profiles in Scan-to-BIM processes, there is no end-to-end approach that effectively addresses these challenges while producing detailed, geometrically accurate, and semantically rich 3D models that preserve critical dimensional parameters like those standardized in EC 3-1-1 notation. This contribution aims to close this identified gap by introducing an automated, tailored method that combines multi-objective optimization for fitting standardized profiles with robust orientation estimation techniques, generating an accurate procedural, partially parametric 3D model from semantically enriched point cloud data without additional user input.

3. METHOD

The presented method is an automated end-to-end method, covering all relevant aspects from semantically segmented point cloud to a semantically rich 3D model. In the following, the relevant steps are introduced individually.

3.1 Input Data and Semantic Segmentation

The raw input point cloud consists of points in 3D space describing the surfaces visible to the scanning equipment. Depending on the scanning system used, this data may include color information and scanning intensities, among others. In the first step, semantic classes need to be separated. This is a time-consuming and arduous task if performed manually. Recent developments in machine learning have shown convincing results in automating this task, as introduced in Section 2.2. The input data is assumed to be available in the form of a semantically segmented point cloud in this work. Nonetheless, the step of *semantic segmentation* is mentioned in Figure 2 (dashed outline) for the sake of completeness in the overall process. The specific data used in the presented work was segmented leveraging synthetic, model-based data. For an in-depth presentation of the dataset, including a detailed report on acquisition and segmentation performance, the reader is referred to (Noichl et al., 2024). In the further processing steps, only the point coordinates and point normals of the semantic class *beam* are utilized.

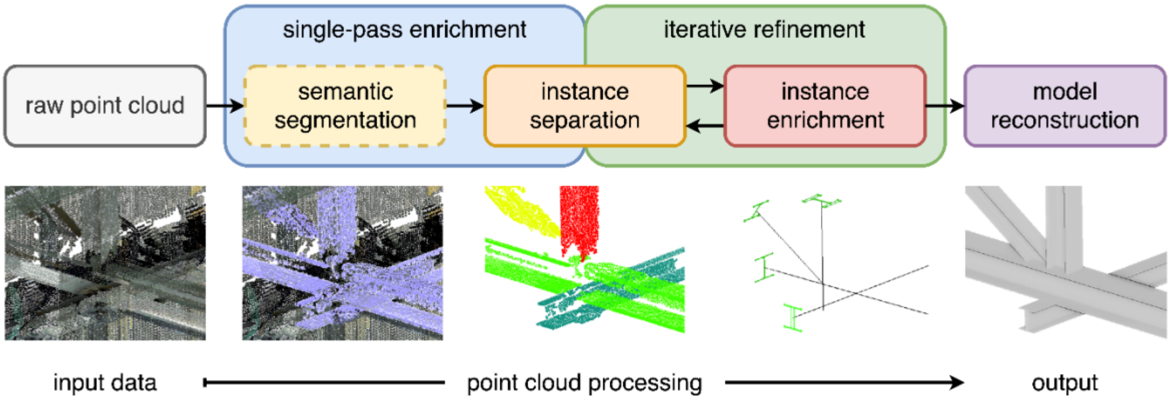


Figure 2: Method overview: from raw point cloud to partially parametric 3D model.

3.2 Point Cloud Processing

The proposed method performs point cloud processing through multiple steps. As depicted in Figure 2, the point cloud processing steps can be separated into single-pass enrichment and iterative refinement steps. These processing steps include specific computation steps, specifically:

1. Planar Patches
2. Local Orientation Supernormal $\overrightarrow{\boldsymbol{s_1}}$
3. Segment Orientation $\overrightarrow{\boldsymbol{s_2}}$ and Point Projection
4. Cross-section Fitting

5. Instance Separation: Advanced Region Growing
6. Instance Refinement: Iterative Aggregation and Skeleton Refinement

In the following, these individual processing steps are introduced in detail.

3.2.1 Planar Patches

The point cloud is clustered into planar patches using RANSAC (RANdom SAmple Consensus, Schnabel et al., 2007) and DBSCAN (Density-Based Spatial Clustering of Applications with Noise, Ester et al., 1996). First, planes are detected in the point cloud using RANSAC iteratively. Inlier points are labeled and removed from the point cloud until no plane with more than the threshold points can be detected. Within these labeled plane clusters, DBSCAN is used to identify coherent planar patches by clustering points based on spatial distribution and local density (see Figure 3). As a result, the point cloud can be processed using these aggregated clusters rather than individual points more efficiently and enriched with the first logic of coherent surface patch points.

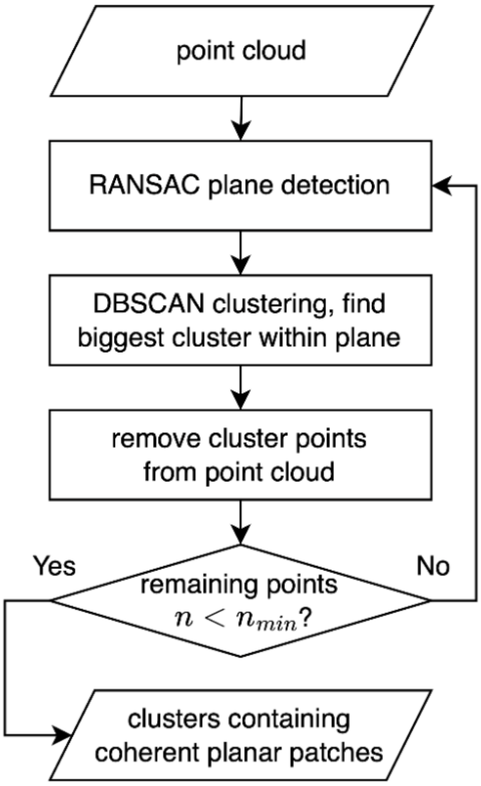


Figure 3: Identification of planar patches in the point cloud using RANSAC and DBSCAN.

This ensures that points are only clustered if they belong to the same planar patch, lying on the same idealistic plane and within spatial proximity, effectively preventing the connection of patches on disjoint beams to the same cluster.

3.2.2 Local Orientation Supernormal $\overrightarrow{s_1}$

Assuming the observed points are located on the surface of an elongated part, their respective normals will point outwards and can be utilized to estimate the orientation of the part. For a group of points ($n_p \geq 2$), the *supernormal* $\overrightarrow{s_1}$, an estimated part orientation feature is calculated based on the point normals (see Figure 4). Initially, the group of points is defined as the inlier points of a local neighborhood with radius r_n. During region growing, all points in a cluster are considered as the input group of points. The points within the group are denoted as $\mathbf{P} = \{\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_n}\}$ with their associated normal vectors $N = \{\overrightarrow{n_1}, \overrightarrow{n_2}, \ldots, \overrightarrow{n_n}\}$, collected in a normals matrix $\mathbf{N}$ with $\overrightarrow{n_i}$ as n rows. In case input points do not possess normal vectors, they need to be estimated based on their respective local neighborhood first. As this step is implemented in numerous libraries and applications, it is out of scope for this work.

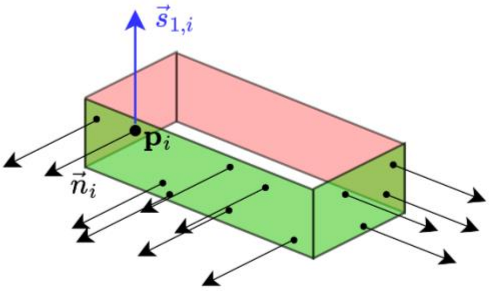


Figure 4: Local orientation feature $\overrightarrow{s_1}$*: Extruded part with rectangular cross-section, partially occluded: green faces are covered by point cloud, red faces are occluded. Dots depict the points of the point cloud, arrows indicate point normal vectors. Exemplary annotated normal* $\overrightarrow{n_i}$ *and supernormal* $\overrightarrow{s_{1,i}}$ *for point* $\boldsymbol{p}_i$*.*

Using all normal vectors in the group, $\overrightarrow{s_1}$ is calculated using SVD (Singular Value Decomposition), describing a direction in space that is the best fit perpendicular vector to all normals in the group. SVD is applied to $\mathbf{N}$ to retrieve

$$\mathbf{N} = \mathbf{U\Sigma V}^T,$$
$$\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3). \tag{1}$$

where $\mathbf{U}$ and $\mathbf{V}$ are the orthogonal matrices obtained from the SVD, $\mathbf{\Sigma}$ constitutes the diagonal matrix with singular values, and $\sigma_1 \geq \sigma_2 \geq \sigma_3$ are the singular values.

$\overrightarrow{s_1}$ can be obtained from the last column of $\mathbf{V}$

$$\overrightarrow{s_1} = \mathbf{V}_3, \tag{2}$$

and is a vector perpendicular to the direction of maximum variance in $\mathbf{N}$, and therefore constitutes a first estimate of part orientation given a group of points on the part's surface, given the that points do not lie on a single plane or multiple parallel planes.

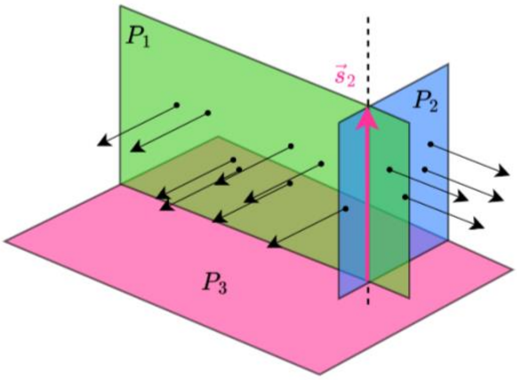


Figure 5: Point cluster: estimated segment orientation $\overrightarrow{s_2}$ *defined as the direction of the intersecting line between the two detected planes* P_1 *and* P_2*, the perpendicular plane* P_3 *has* $\overrightarrow{s_2}$ *as its normal vector.*

If the selected group of points does not belong to parts with the same part orientation, $\overrightarrow{s_1}$ can be misleading. To prevent this, a confidence value c_s is computed along with $\overrightarrow{s_1}$, taking into account the number of points in the supporting set and the variation of normal orientations within N using $\mathbf{\Sigma}$:

$$c_s = \sqrt{|N|} \cdot \frac{\sigma_2}{\sigma_1} \cdot \left(1 - \frac{\sigma_3}{\sigma_2}\right), \tag{3}$$

where $|N|$ denotes the number of points in the supporting set. σ_1 is the smallest singular value, corresponding to the direction of most variance, σ_3 is the largest singular value, corresponding to the direction of least variance, and therefore corresponding to $\overrightarrow{s_1}$. σ_2/σ_1 describes the relation between first and second singular value, which are ideally equal if the normals are evenly distributed, in a plane, σ_3/σ_2 sets the smallest singular value in comparison to the second singular value. The smaller σ_3, the smaller the reduction of c_s by $(1 - \sigma_3/\sigma_2)$. The higher the value of c_s, the more confidence can be attributed to $\overrightarrow{s_1}$ in terms of being a well-supported and accurate vector perpendicular to the normals in the supporting group, and the normals in the group depicting a suitable selection in terms of the principle introduced above and depicted in Figure 4.

3.2.3 Segment Orientation $\overrightarrow{s_2}$ and Point Projection

Elongated parts such as steel beams possess cross-sections such as rectangles, C-, L-, and I-shapes. Such cross-sections can be described by combining multiple geometric primitives, typically planes. Therefore, in the following, they are referred to as composite cross-sections. Along with round shapes, mostly present in pipe or duct systems, composite cross-sections represent the most important part shapes in industrial facilities (Agapaki et al., 2018).

The presented method proposes to estimate the orientation of elongated parts with composite cross-sections using RANSAC for plane detection; the underlying principle is depicted in Figure 5. RANSAC is applied to find the most prominent plane in the set and remove the inlier points from the point cloud. Plane parameters are stored, and RANSAC is repeated until a second plane is found close to perpendicular to the first, within a pre-defined threshold θ (see Figure 6). As RANSAC is a model-based method that is robust to outliers and, therefore, occlusions and scanning noise, the resulting intersecting line between those two planes is chosen to identify part orientation $\overrightarrow{s_2}$.

After the orientation $\overrightarrow{s_2}$ has been identified, all points of the cluster are projected onto an idealistic plane P_3 (see Figure 5), defined by segment orientation feature $\overrightarrow{s_2}$. The projected points are finally aligned to the 2D coordinate system through rotation by the intersecting angle of the most prominent plane P_1 with the x-axis. With the projected representation of points, a 2D center of gravity can be estimated using the mean coordinate values and used for further processing of the point cluster for cross-section fitting.

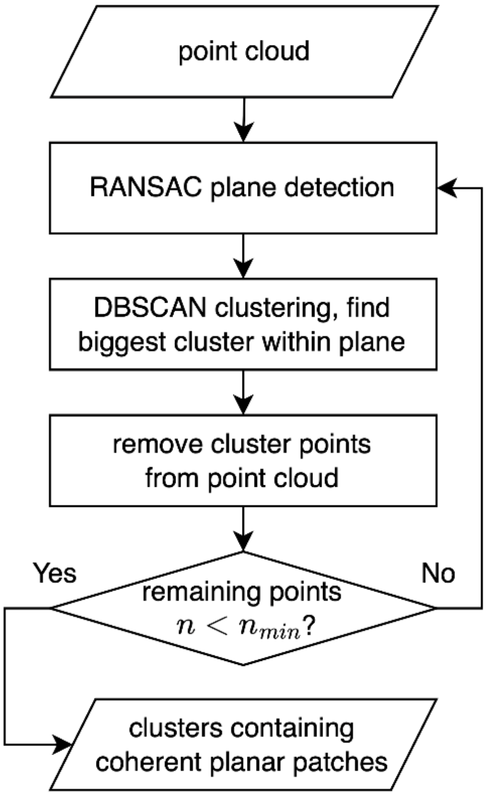
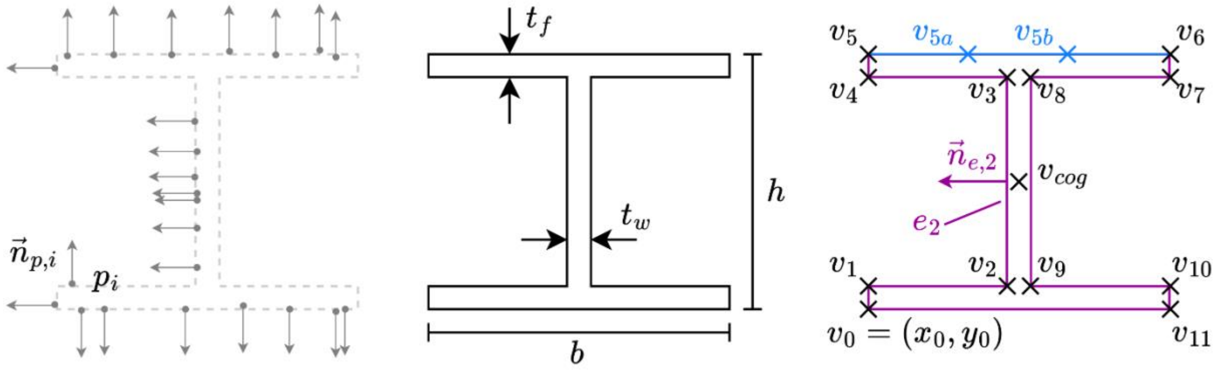


Figure 6: Orientation estimation in point clusters by intersection of detected planes.

3.2.4 Cross-section Fitting

Following instance separation and refinement, the projected and axis-aligned points (see Figure 7a) are used to fit a parametric polygon (see Figure 7b,c) representing an I-shaped steel profile cross-section. Across national standards (e.g., AISC, EC 3-1-1: Figure 1), I-shaped cross-sections are defined by the parameter set $\boldsymbol{\theta} = \{t_f, t_w, b, h\}$, where t_f is the flange thickness, t_w the web thickness, b the flange width, and h the section depth. Edge resolution is controlled by a pre-defined parameter $l_{\max}$ that defines the threshold for edge subdivision: For any polygon derived from a set of shape parameters $\boldsymbol{\theta}$, edges that exceed $l_{\max}$ are subdivided into $n = \lceil l_{\text{edge}}/l_{\max} \rceil$ edges, as depicted in Figure 7c. The location of the cross-section is controlled by one of the polygon vertices, here chosen as $v_0 = (x_0, y_0)$. As depicted in Figure 1, the exact description of the cross-section requires fillet radius r. To reduce computational effort, the parametric polygon used to identify the correct cross-section is defined without fillets by line segments only.



(a) (b) (c)

Figure 7: Projected point cluster with points p_i with point normals $\overrightarrow{n_{p,i}}$ of a partially occluded H-beam (a), H-beam with size parameters $\boldsymbol{\theta}$ (b) and control vertices v_i to define the location v_0 and shape of the cross-section polygon with edge normals, exemplary depicted with $\overrightarrow{n_{e,2}}$ for e_2. Polygon edges can be subdivided into sub-edges of equal length to increase evaluation precision, exemplarily shown with inserted polygon vertices v_{5a}, v_{5b} (c).

The optimal cross-section shape and location are determined using a $(\mu + \lambda)$-Evolution Strategy (Bäck & Schwefel, 1993; Beyer & Schwefel, 2002). Each solution candidate is defined by $\{i, x_0, y_0\}$, where i indexes a standardized parameter set $\boldsymbol{\theta}_i$, and (x_0, y_0) define the polygon's reference point. To evaluate the quality of an individual solution, the parameters are retrieved by index and used to define the polygon and compare it to the projected and axis-aligned points of the beam instance. In a similar approach introduced by Mafipour et al. (2023), a single cost function was used to consider active edges of the polygon and angular constraints between edges to be used in PSO (Particle Swarm Optimization). The challenge for steel cross-section differs from this due to fewer degrees of freedom in the cross-section shape, while the underlying data is significantly more challenging due to significant noise, due to smaller scale, and heavily occluded cross-sections in complex industrial scenes.

Three objective functions are defined to take these considerations into account: The first objective is the logarithmic distance $d_{\log}$ (Equation 4), which is the most essential, calculating the distance between the point cloud and the polygon. The logarithmic calculation allows for the prioritization of close distances and increases robustness against outliers in the point cloud, introduced through errors in segmentation or measurement noise:

$$d_{\log} = \frac{1}{N}\sum_{i=1}^{N} \log\left(\min_j d_{ij}\right), \tag{4}$$

where N is the number of data points and d_{ij} is the closest distance from point i to polygon edge j. The second objective is the relative active edge length $l_{\text{a,rel}}$, indicating the relative share of the polygon close to the data points:

$$l_{\text{a,rel}} = \frac{\sum_{k=1}^{M} a_k \cdot l_k}{\sum_{k=1}^{M} l_k}, \tag{5}$$

where M is the number of edges in the polygon ($M \geq 12$, depending on $\boldsymbol{\theta}$ and $l_{\max}$), a_k is the activity of each edge k (0 or 1), and l_k is the length of edge k. An edge is considered active if there is a point in the point cloud that is closer to edge k than any other edge. Computing the relative active length ensures that all sizes of cross-sections are treated equally. To improve the precision of $l_{\mathrm{a,rel}}$, polygon edges can be subdivided using a maximum edge length threshold l_{max}, as depicted in Figure 7c. Edge normals are adopted for each resulting edge as the original normal vector of the edge.

The third objective is the normal orientation similarity φ_n, describing the similarity of point normal directions and their matching edge normals:

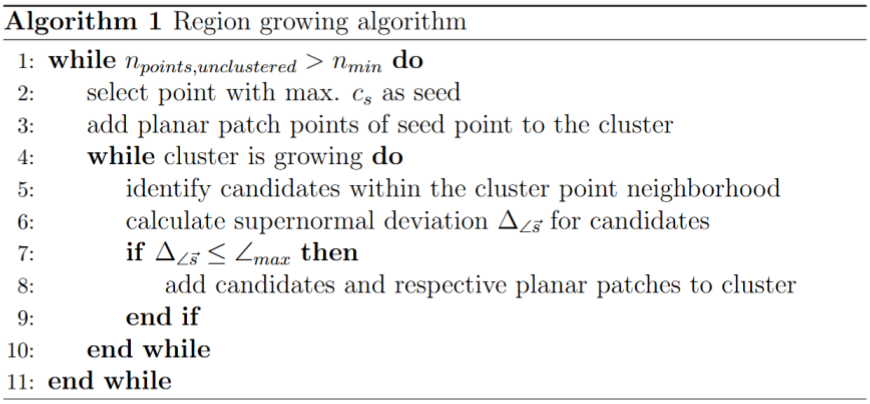
$$\varphi_n = \sum_{e \in E_{\mathrm{active}}} \sum_{i \in P_e} \cos(\overrightarrow{n_e}, \overrightarrow{n_{p,i}}), \tag{6}$$

where E_{active} is the set of active edges, P_e is the set of points associated with edge e, $\overrightarrow{n_e}$ is the normal vector of edge e and $\overrightarrow{n_{p,i}}$ is the normal vector of point i. Evaluating point coordinates alone in heavily occluded cross-sections can lead to erroneous results for parts of which only one surface is visible, such as the right surface of the web depicted in Figure 7. The normal orientation similarity penalizes cases where points are assigned to an edge facing the opposite direction than the point normal.

Due to varying instance segment sizes, ranging from hundreds to thousands of points, a downsampling step using k-means clustering (Lloyd, 1982) limits the input to a computationally manageable, pre-defined size while preserving the point cloud's geometric characteristics. Each centroid in the downsampled point cloud is assigned a weight corresponding to the number of original points it represents. These weights are incorporated into all three objective functions, ensuring that dense regions of the original point cloud maintain their relative importance in the optimization process.

3.2.5 Instance Separation: Advanced Region Growing

To separate instance clusters depicting single beam objects, the introduced methods are applied in single-pass and iterative processing steps. Iterative refinement is performed until the resulting cluster of the region-growing process remains unchanged for one iteration. The method is explained in Algorithm 1. Region growing in the presented method should result in over-segmentation rather than under-segmentation, because part limits need to be respected in the result of region growing: parts that are separated in multiple instance clusters can be aggregated in the following step, but wrongly joined clusters cannot be separated downstream.

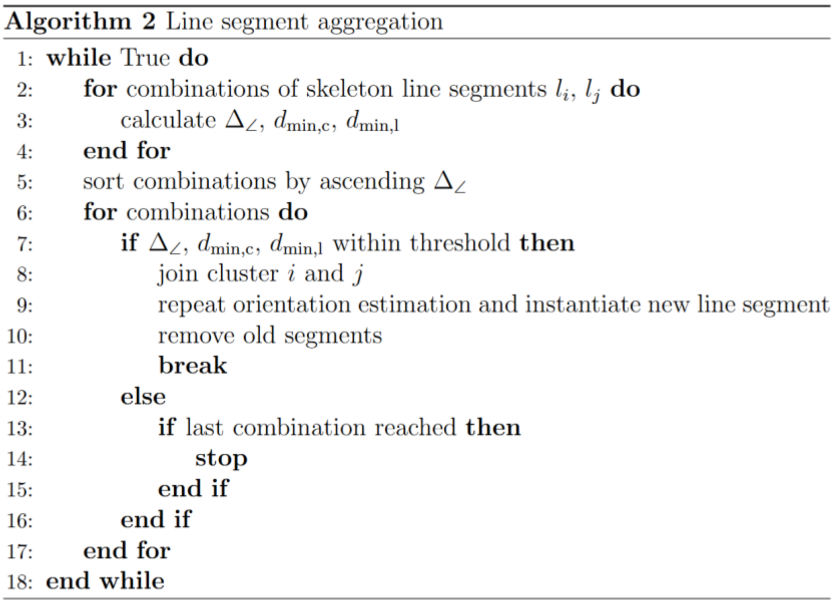

Algorithm 1 Region growing algorithm

```
while $n_{points,unclustered} > n_{min}$ do
    select point with max. $c_s$ as seed
    add planar patch points of seed point to the cluster
    while cluster is growing do
        identify candidates within the cluster point neighborhood
        calculate supernormal deviation $\Delta_{\angle\vec{s}}$ for candidates
        if $\Delta_{\angle\vec{s}} \leq \angle_{max}$ then
            add candidates and respective planar patches to cluster
        end if
    end while
end while
```

Depending on occlusions and noise in the input point cloud, using region growing as a standalone method for clustering usually leads to over-segmentation; complete instances can often not be identified as such and need to be aggregated in another step (see Section 3.2.6). If ground truth information is available as point-wise labels, this can be quantified and controlled by observing performance set metrics like *precision* and *recall*. The application and meaning of these metrics in the context of the presented method are important for the experiment and explained in Section 4.3.

3.2.6 Instance Refinement: Iterative Aggregation and Skeleton Refinement

After initial clusters are identified in the region growing step, they are used as input for instance refinement. In this step, the cluster orientation is estimated using the method introduced in Section 3.2.3, resulting in a more robust estimation of part orientation. The results are used to instantiate parts of the skeleton: the known location of the cluster point coordinates, the projected center of gravity in 2D, and the obtained $\overrightarrow{s_2}$ are used to define the centerline of the segment and project the cluster points to that direction vector. By identifying the first and last point after projection to this direction vector, a line segment can be defined as part of the skeleton of the beam system.



Algorithm 2 Line segment aggregation

```
while True do
    for combinations of skeleton line segments $l_i$, $l_j$ do
        calculate $\Delta_\angle$, $d_{\mathrm{min,c}}$, $d_{\mathrm{min,l}}$
    end for
    sort combinations by ascending $\Delta_\angle$
    for combinations do
        if $\Delta_\angle$, $d_{\mathrm{min,c}}$, $d_{\mathrm{min,l}}$ within threshold then
            join cluster $i$ and $j$
            repeat orientation estimation and instantiate new line segment
            remove old segments
            break
        else
            if last combination reached then
                stop
            end if
        end if
    end for
end while
```

After the skeleton is instantiated, angles and distances between all parts are calculated. This includes the angular difference between part vectors, the minimum distance between two skew vectors, and the minimum distances between the start and endpoints. If these values are within the pre-defined thresholds, two skeleton line segments are joined, the old clusters and line segments are removed, and the estimation of segment orientation $\overrightarrow{s_2}$ and point projection are repeated for the new cluster. This procedure is explained in sequence and pseudo-code in Algorithm 2, and the mentioned variables are explained visually in Figure 8.

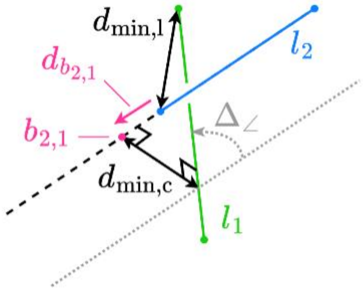


Figure 8: Relationship between skeleton line segments l_1 and l_2: intersecting angle $\Delta_\angle$, minimum lengthwise distance of endpoints $d_{min,l}$, minimum crosswise distance between warped vectors $d_{min,c}$. These parameters are checked for a potential extension of l_2 to bridge point $b_{2,1}$ with by distance $d_{b_{2,1}}$.

Similar to the region growing algorithm (see Algorithm 1), this iterative method is stopped if an iteration does not yield any change in the skeleton line segments. By considering three independent parameters for joining segments, this method bridges gaps caused by significant occlusions, using the plane-based segment orientation estimation of $\overrightarrow{s_2}$ as its main feature.

Occlusions on steel beams can occur on free length as well as in joint areas and can by no means be assumed to be consistent along the length of a beam. The presented method bridges gaps in the free length of beams by the segment aggregation method described above, joint areas need to be considered in the final step of skeleton refinement. To ensure that no gaps persist in model reconstruction, line segments close to joining onto other line segments are extended to their bridge points, defined by the location of the minimum distance between two warped vectors, as depicted in Figure 8. The underlying algorithm is similar to Algorithm 2. However, threshold $d_{\mathrm{min,l}}$ is replaced with a threshold value for extending l_i, d_l does not have to be considered.

3.3 Model Reconstruction

In the last step of the presented method, all obtained information is used as input data to reconstruct the 3D model of the steel structure. The sequence of steps is depicted in Figure 9. Using Blender[1] and Bonsai[2], the beam type can be used directly to retrieve the required cross-section from a standardized catalog. Using the determined length of the respective line segment, an extrusion is performed to create the beam geometry in its 3D representation. The extruded profile is finally placed in its target location by rotation and translation. The necessary parameters are obtained in the orientation estimation step (see Section 3.2.3).

Generating structural 3D models with detailed beam connections is already solved through existing proprietary and open-source applications, and is therefore outside the scope of our method. Furthermore, model reconstruction with the required level of detail for connectors requires high-resolution local information. However, such depth of information is usually not captured, and to integrate such aspects would lead to unnecessary ambiguity. On the contrary, the presented basic parametric model reconstruction at LOD 300 (BIMForum, 2024) is based on precise, automatically retrieved information.

Working with Bonsai in Blender in the described way allows the export of the generated 3D model of the steel structure in two distinct ways. First, the model is exported to the open IFC data exchange format, which contains rich semantic information such as detailed information on steel profile cross-sections[3]. Generally, this format enables seamless further processing in other software applications for further analysis, planning, or quantity takeoff. Second, the geometric representation of each beam and the entire structure is exported to a 3D mesh representation for geometric evaluation of the resulting geometry and comparison to the input point cloud.

4. EXPERIMENTS AND RESULTS

The effectiveness of the proposed method is demonstrated through comprehensive experiments on laser-scanning point cloud data from an industrial environment. This section presents implementation details first, followed by a description of the dataset and the methodology for establishing ground truth. Then, the relevant metrics for evaluation are introduced, including their mathematical formulation and interpretation. Two experiments validate our approach: first, a detailed step-by-step analysis on a subset of the dataset illustrates the internal workings of the method, while a second experiment evaluates the method's performance on the complete dataset to demonstrate its scalability and robustness in real-world scenarios.

4.1 Implementation

The presented method is implemented using several specialized open-source libraries, namely Open3D[4], DEAP[5], and Blender with Bonsai. Using the authors' implementation, several experiments are conducted to investigate the functionality of the presented method. The authors' implementation is made available at https://github.com/fnoi/pc2beam; no proprietary, commercial software was used in this project. We refer the reader to this repository for comprehensive information on all utilized libraries beyond the above-mentioned examples.

Several parameters define the method's performance, as noted throughout Section 3. They are collected with the respective processing steps in Table 2. Some parameters are not explicitly mentioned in this collection as they depend solely on properties of the input point cloud and do not require tuning based on geometry, such as RANSAC and DBSCAN parameters.

---

[1] https://blender.org, v.4.2.2

[2] https://bonsaibim.org, v.0.8.0

[3] http://128.199.55.109/IFC/RELEASE/IFC4x3/HTML/lexical/IfcIShapeProfileDef.htm

[4] https://www.open3d.org/, v.0.18

[5] https://github.com/deap, v.1.4.1

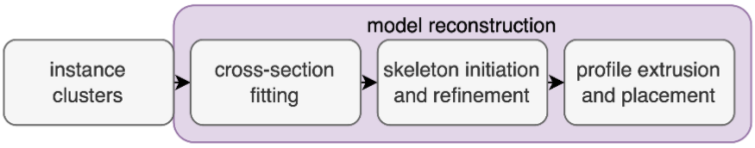


Figure 9: 3D model reconstruction method overview.

Table 2: Collected relevant parameters per computation step.

Processing step	Parameter	Description
local orientation supernormal $\vec{s_1}$	r_n	neighborhood radius
instance separation	$\angle_{\max}$	max. angular orientation $(\vec{s_1}, \vec{s_2})$ deviation
iterative aggregation	$\Delta_{\angle,max}$	max. angular deviation $(\vec{s_2})$
	$d_{\max,\mathrm{c}}$	max. crosswise distance
	$d_{\max,\mathrm{l}}$	max. lengthwise distance
instance refinement	$d_{\mathrm{b},\max}$	max. distance to bridge point

4.2 Dataset

The dataset used for the presented experiment depicts the steel roof structure of an industrial building. It was captured using a Faro Focus 3D terrestrial laser scanner and includes $7.8 \cdot 10^6$ points classified as beams. A detailed report on the dataset, including scanning parameters and overall class distributions, is reported in (Noichl et al., 2024).

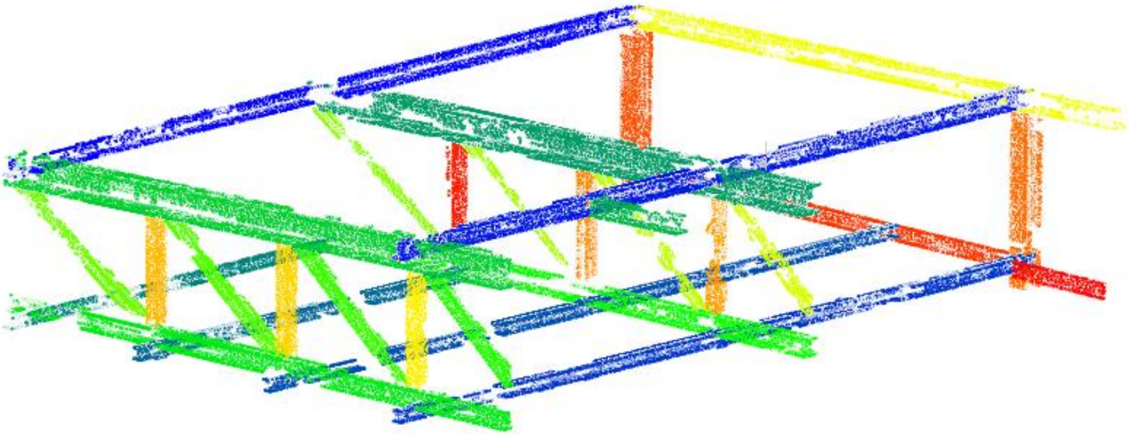


Figure 10: Steel roof structure point cloud, ground truth beam instances color-coded.

Ground truth is created manually to provide a basis for the subsequent calculation of performance metrics. The annotation of instances is performed by hand within the *beam* point cloud, as depicted in Figure 10. A total of 30 beam instances are separated, and for each beam, the orientation is manually estimated by adjusting the camera angle until the best possible projection direction is identified. Further, the point cluster of each beam is evaluated individually to provide insight into variability in the data and effects of occlusion. The number of points within local neighborhoods is evaluated to compare the characteristics of beam instances in terms of point density. As reported in Figure 11, the variability within the used dataset is significant: instance cluster sizes range between 1'213 and 58'232 points, the mean number of points within the local neighborhood with radius $r = 0.1$m ranges between 124 and 480 points. The smallest instances also have the least local point density (see Figure 11 left): with increasing mean number of points in the local neighborhood, the standard deviation also increases (see Figure 11 right), which indicates inhomogeneous point densities caused by occlusions in the laser scan. The standardized cross-section types of the investigated beams in the dataset vary between wide-flange beams HEA340 (largest cross-section) and HEA140 (smallest cross-section).

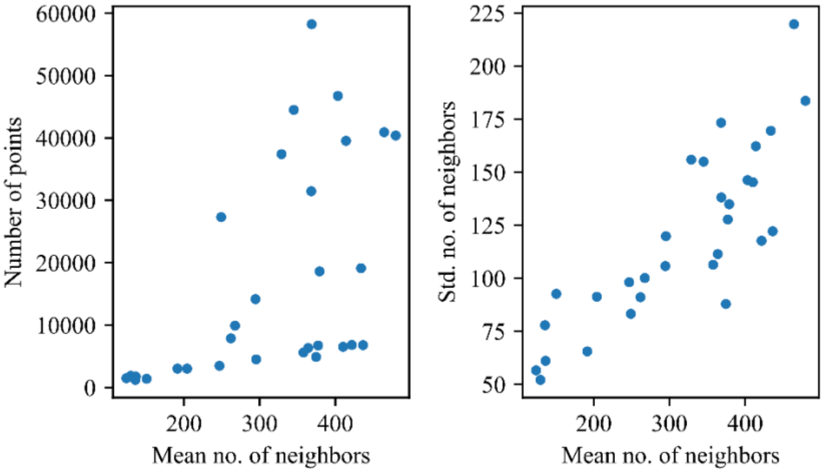


Figure 11: Beam instance characteristics in point cloud dataset with local neighborhood radius $r = 0.1m$. Left: total number of points vs. mean number of points in local neighborhood, right: standard deviation vs. mean number of points in local neighborhood.

4.3 Evaluation Metrics

In the following experiments, the outcomes of individual steps and the final results are evaluated quantitatively. This is done using geometric and classification metrics. Geometric deviations are measured in terms of point-mesh distance and angular deviation. The point-mesh distance d_{pm} describes the minimum distance between a point and a mesh surface. This calculation is a standard feature of state-of-the-art applications such as CloudCompare[6] and not discussed in detail. The angular deviation θ_Δ between two vectors in space can be calculated as follows:

$$\theta_\Delta = \arccos\left(\frac{\vec{a}\cdot\vec{b}}{|\vec{a}|\cdot|\vec{b}|}\right), \tag{7}$$

Where $\vec{a}\cdot\vec{b}$ is the dot product of the two vectors and $|\vec{a}|$ and $|\vec{b}|$ represent the magnitudes of the vectors.

The purity metric is utilized to assess homogeneity in the initially over-segmented clusters. As denoted in Equation 8, it reflects the mean consistency of labels within predicted clusters.

$$\text{purity} = \frac{1}{n}\sum_{i=1}^{k} \max_j |c_i \cap t_j|, \tag{8}$$

where n is the total number of points, k is the number of predicted instances, c_i is the i-th predicted instance, and t_j is the j-th ground truth label (Manning et al., 2008).

To quantify general clustering performance, the conventional classification metrics of precision (P), recall (R), and Intersection over Union (IoU), and an aggregated mean IoU (mIoU) are evaluated. To consider imbalanced segment sizes, the weighted mean IoU (wmIoU) considers the number of points in each segment. These metrics can be evaluated in a controlled environment with known ground truth:

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \tag{9}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \tag{10}$$

---

[6] https://www.danielgm.net/cc/, v.2.13.2

$$\mathrm{IoU_i} = \frac{\mathrm{TP_i}}{\mathrm{TP_i}+\mathrm{FP_i}+\mathrm{FN_i}}, \tag{11}$$

$$\mathrm{mIoU} = \frac{1}{\mathrm{C}}\sum_{\mathrm{i}=1}^{\mathrm{C}} \mathrm{IoU_i},$$

$$\mathrm{wmIoU} = \frac{\sum_{i=1}^{C} n_i \cdot IoU_i}{\sum_{i=1}^{C} n_i}, \tag{12}$$

where TP denotes true positives, FP false positives, FN false negatives; n_i denotes the number of points in segment i; and subscript i attributes these metrics to beam instances; C denotes the total number of segments.

Instance segmentation presents a particular challenge for evaluation, as the relationship between predicted and ground truth instances is not a $1:1$ mapping but rather an $m:n$ correspondence. To address this, a systematic approach is employed. First, an IoU value is calculated for each possible pairing of predicted and ground truth instances. This process results in a $m \times n$ matrix of IoU scores, where m is the number of predicted instances and n is the number of ground truth instances. To determine the optimal matching between predicted and ground truth instances, the Hungarian algorithm (also known as the Munkres algorithm) is applied to the IoU matrix. This algorithm determines the $1:1$ assignment that maximizes the overall sum of IoU values and thus mIoU. The resulting matching strategy provides a basis for calculating precision, recall, and other instance-level metrics. Matched pairs are considered TP, unmatched predictions are FP, and unmatched ground truth instances are FN.

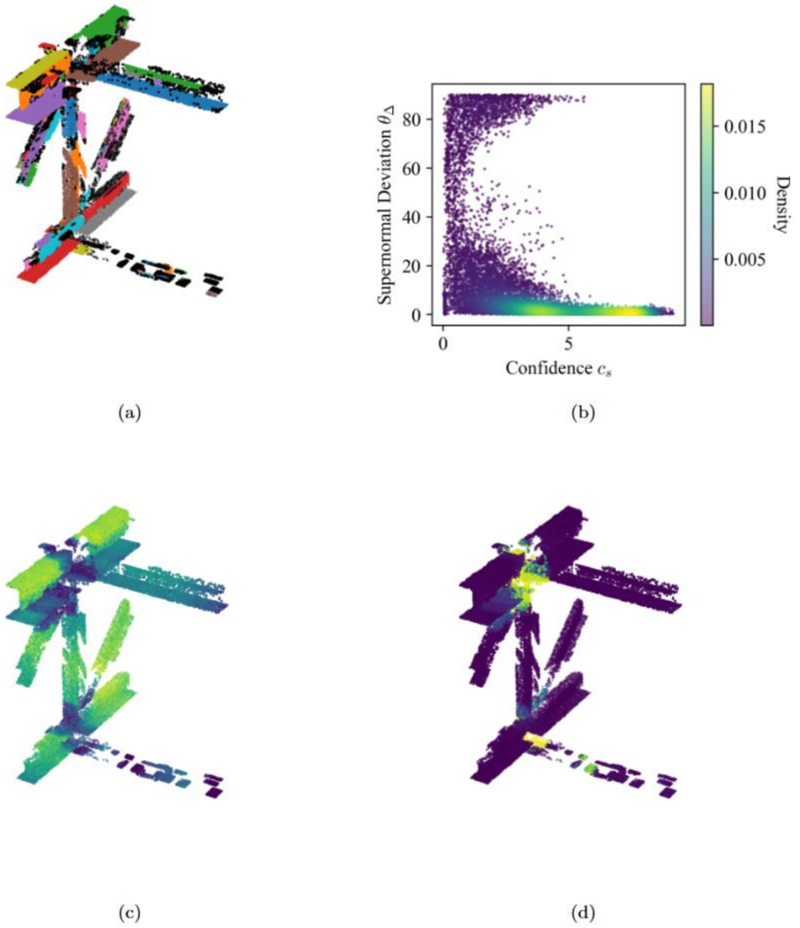


Figure 12: Evaluation of intermediate results in the initial region: planar patches color-coded, unclustered points black (a); analysis of supernormal $\overrightarrow{s_1}$ quality in the initial region (b-d): correlation between supernormal confidence c_s and deviation between supernormal and ground truth orientation θ_Δ, density of scatter points added to aid visualization (b), supernormal confidence in the initial region (c), deviation between supernormal orientation $\overrightarrow{s_1}$ and part orientation ground truth (d).

To consider the segment size imbalance in the reported metrics, the performance of instance segmentation is evaluated using the weighted mean Intersection over Union (wmIoU) or Aggregated Jaccard Index (AJI) (Kumar et al., 2017).

4.4 Experiment 1: Processing Step Evaluation

In the first experiment, the performance of individual processing steps is evaluated as each step influences the final result of the presented Scan-to-BIM method. An initial point cloud region is selected and annotated to find suitable parameters for the presented method (see Table 2).The initial region chosen for this experiment consists of 7 beams and a total of 26'897 points (see Figure 12). Table 3 collects the parameters chosen for this region. Note that the parameters for planar patch identification are fixed values and not investigated in the presented experiment.

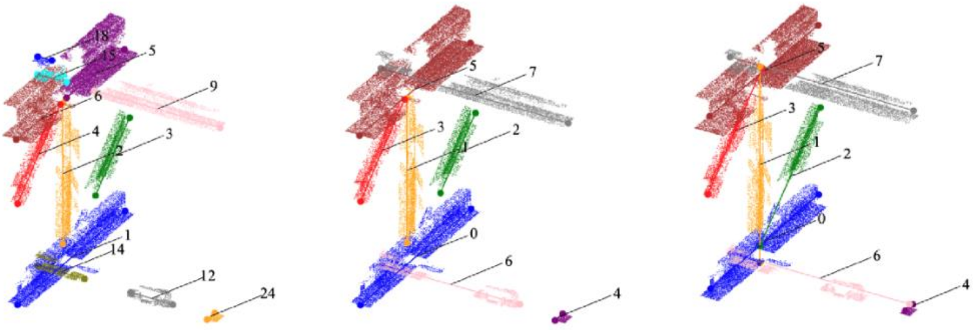
Table 3: Chosen parameter values for processing of the initial region.

Parameter	r_{n}	$\angle_{\max}$	$\Delta_{\angle,\max}$	$d_{\max,c}$	$d_{\max,l}$	$d_{b,\max}$
Value	0.05 m	10°	15°	0.1 m	0.1 m	0.3 m

Planar patch clustering leads to a total of 106 patches with a mean size of 251 points. About 20% of points remain unclustered, as depicted in Figure 12a. As per Equation 8, the resulting cluster purity is 98.9%, which means that only 1.1% of clustered points differ from the most common label in the clusters. At this early stage, this is crucial: if planar patches include points of multiple beam instances, this error will persist through all computation steps in the presented method.

The initial calculation of the local orientation supernormal $\overrightarrow{s_1}$ is performed for each point in the point cloud based on its neighbors within a spherical neighborhood with radius r_{n}. This feature is stored as a 3D vector for each point, so deviations can be evaluated against the manually annotated ground truth on the point level. Selected results are depicted in Figure 12. These results show how $\overrightarrow{s_1}$ works in principle: Figure 12b shows that for the majority of points, $\overrightarrow{s_1}$ is a reasonable estimation of the part orientation (deviation from ground truth: mean deviation 7.7°, median 0.85°); the highest density of points in the plot is found for high confidence c_s and low deviation θ_Δ. In Figure 12c, c_s is color-coded for each point in the cloud, revealing the critical areas for supernormal estimation using spherical neighborhood for joints between multiple beams and for heavily occluded sections. A similar pattern can be observed in Figure 12d, where the points are color-coded according to θ_Δ, further supporting the confidence value of c_s as an indicator of the quality of the local orientation feature $\overrightarrow{s_1}$.

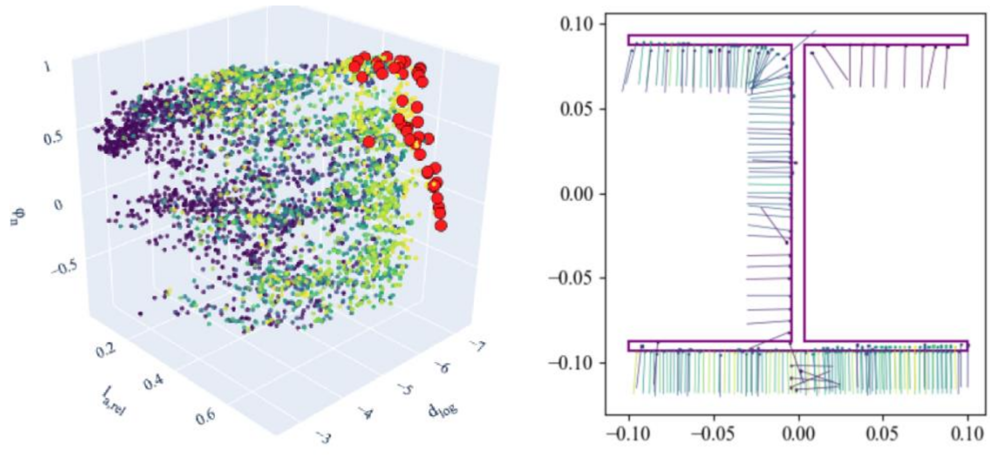
With the chosen parameters, region growing leads to wmIoU of 38.9%, while a high mean cluster purity can be retained with 99.5% (worst cluster purity = 90.0%). The color-coded initial clusters are depicted in Figure 13a. At this stage, the point cloud is clearly oversegmented. Similar to the findings for the planar patches, the challenging areas are the joints between multiple beams and the highly occluded beam in the lower right section.



(a) (b) (c)

Figure 13: Intermediate outcomes and final result of instance segmentation: (a) after region growing, (b) after skeleton line segment aggregation, (c) final skeleton after refinement.

The estimation of segment orientation $\overrightarrow{s_2}$ is performed on the initial clusters to find cluster-wide orientation and initiate the skeleton, as depicted in Figure 13a. The orientation of the resulting original line segments in the skeleton deviates from the ground truth by a point-wise average of 0.1° (median 0.07°). These line segments are used to aggregate the skeleton, as depicted in Figure 13b; the updated instance segments amount to a wmIoU of 89.0%, with a slightly decreased mean cluster purity of 98.9% (worst cluster purity = 97.0%), notably improving the value for worst cluster purity. Finally, the line segment skeleton is refined to ensure system connectivity, as depicted in Figure 13c.

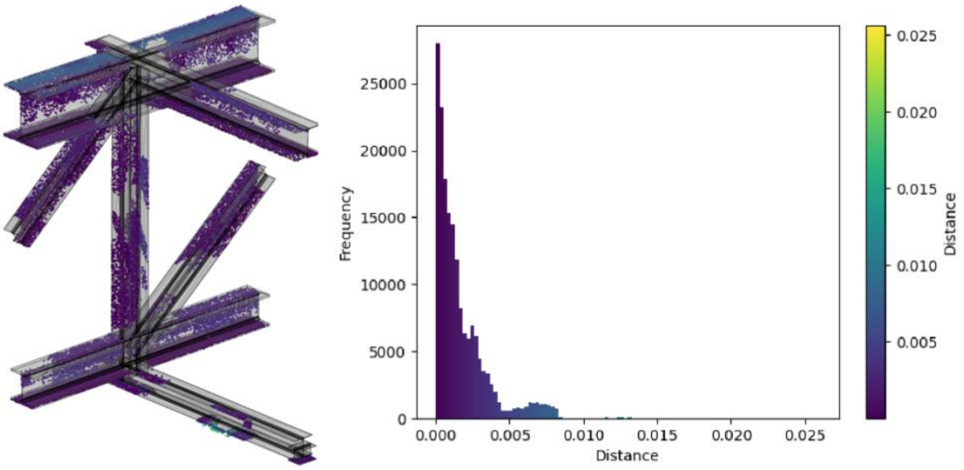


(a) (b)

Figure 14: Exemplary results of cross-section fitting using multi-objective optimization:(a) objective function values for all individuals, color-coded by generation, red circles indicate Pareto front solutions, (b) cross-section polygon and points of a largely occluded beam segment projected to 2D, lines indicate point normals, color-coding indicates point weights.

The aggregated information of the line segment skeleton is used as input for multi-objective optimization to find the right steel profile cross-section type and location. Figure 14 contains two typical results of this process, shown as the evaluated collection of all investigated solutions and the Pareto front, as well as the parametric I-shaped cross-section polygon for a selected H-beam.

After complete model generation, the resulting beam geometries are exported as a high-quality IFC model and mesh representation. The mesh is further used to investigate the quality of the resulting model. Figure 15 shows the point cloud and complete resulting model for the initial region. The overall deviation between the point cloud and model is 1.7 mm (median 1.1 mm).



(a) (b)

Figure 15: Final model output and point deviation: (a) point cloud and beam geometries, (b) point to mesh distance histogram, color-coded using the same scale.

4.5 Experiment 2: Full Dataset Experiment

To investigate the efficacy of the method presented in this paper, it is finally applied to the full dataset with the same parameter setup as in the previous experiments, resulting in the reconstructed model depicted in Figure 16. In this, 24 out of a total of 30 or 80% of beam instances are reconstructed successfully (see Figure 10 for reference). The missing beams are lost due to one critical problem: Overall low numbers of points in the segments lead to insufficient points on intersecting planes for determining orientation within an established cluster; these instances are therefore neglected (see Figure 11, bottom left). For the successfully identified instances, the mean deviation between the point cloud and the reconstructed model surface is 4.1 mm (median 2.8 mm). The missing instances increase the mean deviation between the full input point cloud and reconstructed model surface to 20 mm (median 3 mm). Between ground truth beam orientation and orientation of the beam objects in the reconstructed model, the mean angular deviation remains at 0.1°. For a single beam in the dataset, estimation of longitudinal rotation failed – as depicted in Figure 16, bottom right, one beam is rotated by 90°.

All semantic information on the beam cross-section, including standardized cross-section profile names and parameters, is finally stored in an IFC file to enable the best interoperability for the resulting models (see Figure 17).

Figure 16: Resulting IFC model for the full dataset. Visualized using IfcOpenShell viewer[7].

4.6 Discussion

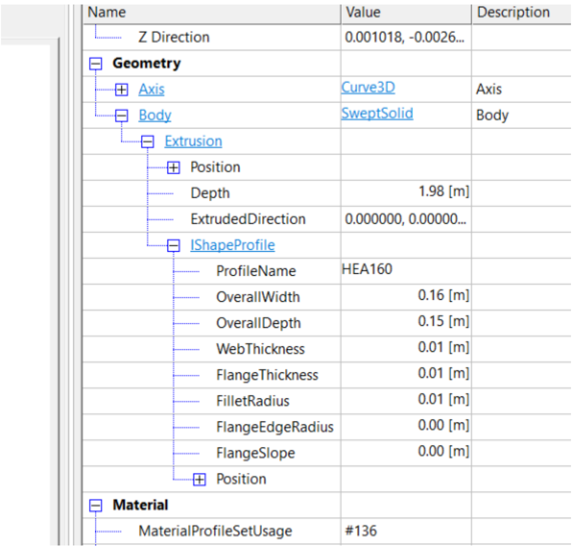
The results of these experiments exhibit several strengths and limitations of the presented method. Section 4.4 shows that the local orientation supernormal feature $\overrightarrow{s_1}$ is able to estimate the part orientation based on a local neighborhood directly. However, in sections of the scene with complex geometry, such as beam joints, $\overrightarrow{s_1}$ does not produce reliable estimations. The presented confidence measure of c_s identifies points with weak support for $\overrightarrow{s_1}$ in the further process and mitigates the risk of erroneous decisions based on this feature. Orientation estimation on the cluster or segment level for $\overrightarrow{s_2}$ using intersecting planes identified using RANSAC leads to highly accurate results. The precision of $\overrightarrow{s_2}$ is a crucial prerequisite for cross-section fitting using multi-objective optimization, as it defines the input data both in terms of direction and longitudinal alignment of the point projection. The defined objective functions effectively leverage the limited information carried in highly occluded beam segments and return standardized profile cross-sections.

The second experiment in Section 4.5 tests the method's abilities and reveals several limitations. Due to very low local point density and occlusions leading to a low total number of points in several segments, some beams cannot be detected; others might not be connected correctly. The presented instance segmentation method is a sequential heuristic that is relatively inflexible. It is a very good solution for the investigated dataset since hardly any training

---

[7] https://view.ifcopenshell.org, accessed 20-12-2024.

data is available for industrial buildings. For a more generalizable approach, using learning-based, unsupervised methods could be advantageous. A single beam's misidentified longitudinal orientation in this experiment exposed an additional type of error not yet accounted for in the methodology.



Name	Value	Description
Z Direction	0.001018, -0.0026...	
Geometry		
Axis	Curve3D	Axis
Body	SweptSolid	Body
Extrusion		
Position		
Depth	1.98 [m]	
ExtrudedDirection	0.000000, 0.00000...	
IShapeProfile		
ProfileName	HEA160	
OverallWidth	0.16 [m]	
OverallDepth	0.15 [m]	
WebThickness	0.01 [m]	
FlangeThickness	0.01 [m]	
FilletRadius	0.01 [m]	
FlangeEdgeRadius	0.00 [m]	
FlangeSlope	0.00 [m]	
Position		
Material		
MaterialProfileSetUsage	#136	

Figure 17: Exemplary section of beam cross-section information of a beam instance with HEA160 profile cross-section stored in the generated IFC model, retrieved using KITModelViewer[8]*.*

5. CONCLUSIONS AND OUTLOOK

5.1 Contributions

The presented methodology provides two main contributions: Firstly, a robust method to estimate the orientation of elongated parts on the local and segment levels and a sequential heuristic to segment such elements in point clouds: Given a point cloud that partially covers two non-parallel surfaces of a steel beam – or an elongated part in general – the method can first estimate the rough part direction to initiate instance segmentation. Once a segment has been established, the method can precisely estimate part orientation using the presented plane intersection method. Secondly, the presented method describes an end-to-end approach for Scan-to-BIM that comprises various processing steps starting from a semantically segmented point cloud, resulting in a precise, procedural and semantically rich 3D model of the structure, including export to the open standard IFC format for best interoperability in the domain. Both contributions are demonstrated and quantitatively evaluated on a manually annotated laser scanning point cloud dataset depicting the roof structure of an industrial building.

5.2 Scope and Limitations

This study focuses on I-shaped beams, mainly due to limitations in available experimental data. While the presented approach uses extrusions along straight paths for beam modeling, it is important to note that this method does not capture the full detail of reality as-is. Real-world beams may exhibit variations such as sag or slight deviations from perfect straightness. Such deviations are relevant for observations of structural behavior and deformation monitoring and out of scope for the presented work, which aims to contribute to the automation of Scan-to-BIM in general. Detailed joint modeling is intentionally excluded from this study, as it often requires specialized software and techniques outside of the focus of this work. Other studies have demonstrated that Terrestrial Laser Scanning (TLS) accuracy is sufficient for investigating features like cable sag (Suchocki et al., 2024). While data acquisition technology is capable of providing detailed information, the level of detail in as-built or as-is models depends on specific requirements and how these are reflected in the chosen model type and

---

[8] https://www.iai.kit.edu/english/4561.php, accessed 20-12-2024.

might require more detailed modeling approaches than purely parametric cross-sections, i.e., by finer granularity in B-Rep (boundary representation) model with additional control points (Rausch & Haas, 2021).

5.3 Extending Applicability and Future Works

The presented method can be enhanced without altering its fundamental structure or functionality. Its applicability could be extended, for example, to cover all types of composite cross-sections by introducing an intermediate cross-section classification step. While the current implementation is limited to I-shaped beams, the method's core strength lies in its capability to segment, enrich, and reconstruct composite cross-sections. To extend this capability to all code-defined cross-section types, an additional processing step is needed: after point projection, object detection or model fitting techniques could be integrated to differentiate between cross-section types (e.g., rectangular, C, and L-shapes). This classification would then inform the subsequent model fitting process, ensuring that suitable parametric polygon shapes are selected from the catalog to initiate the optimization process.

The current approach reconstructs individual beams and their connections to more complex structural systems based primarily on relative placement. This framework could be enhanced with additional structural logic to address more complex scenarios, such as steel frames or connections to other structures or fixed points. It is worth noting that the method contains parameters that require adjustment based on input point cloud characteristics (density, noise levels, occlusion patterns). To improve usability with user input, the approach could be adapted to an incremental workflow similar to that proposed by Szabo et al. (2023). While supervised learning approaches for instance segmentation exist, the lack of available training data for industrial elements makes unsupervised approaches and heuristics, as demonstrated in our method, particularly advantageous.

Beyond extending the method for wider practical application, future work could investigate existing steel beams in greater detail. The current approach identifies steel beam cross-sections based solely on point cloud geometry. While this provides precise geometric estimation, incorporating additional data such as surface reflectivity could help identify beam coatings, further increasing precision and adding valuable information to material inventories.

ACKNOWLEDGEMENTS

The point cloud data used in the presented experiment was provided by Audi AG. We appreciate their support. We further thank Martin Slepicka and Sebastian Esser for the fruitful discussions and implementation directions for the topic of IFC.

REFERENCES

Abreu, N., Pinto, A., Matos, A., & Pires, M. (2023). Procedural Point Cloud Modelling in Scan-to-BIM and Scan-vs-BIM Applications: A Review. ISPRS International Journal of Geo-Information, 12(7), 260. https://doi.org/10.3390/ijgi12070260

Adán, A., Quintana, B., Prieto, S. A., & Bosché, F. (2018). Scan-to-BIM for ‘secondary’ building components. Advanced Engineering Informatics, 37(November 2017), 119–138. https://doi.org/10.1016/j.aei.2018.05.001

Agapaki, E., & Brilakis, I. (2021). Instance Segmentation of Industrial Point Cloud Data. Journal of Computing in Civil Engineering, 35(6), 04021022. https://doi.org/10.1061/(asce)cp.1943-5487.0000972

Agapaki, E., Glyn-Davies, A., Mandoki, S., & Brilakis, I. (2019). CLOI: A Shape Classification Benchmark Dataset for Industrial Facilities. Computing in Civil Engineering, 66–73. https://doi.org/10.1061/9780784482445.009

Agapaki, E., Miatt, G., & Brilakis, I. (2018). Prioritizing object types for modelling existing industrial facilities. Automation in Construction, 96, 211–223. https://doi.org/10.1016/j.autcon.2018.09.011

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., & Savarese, S. (2016). 3D semantic parsing of large-scale indoor spaces Supplementary. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem, 1534–1543. https://doi.org/10.1109/CVPR.2016.170

Bäck, T., & Schwefel, H.-P. (1993). An Overview of Evolutionary Algorithms for Parameter Optimization. Evolutionary Computation, 1, 1–23. https://doi.org/10.1162/evco.1993.1.1.1

Beyer, H.-G., & Schwefel, H.-P. (2002). Evolution strategies - A comprehensive introduction. Natural Computing, 1, 3–52. https://doi.org/10.1023/A:1015059928466

BIMForum. (2024). Level of Development (LOD) Specification. http://bimforum.org/lod/

Borah, B., & Bhattacharyya, D. K. (2004). An Improved Sampling-Based DBSCAN for Large Spatial Databases. International Conference on Intelligent Sensing and Information Processing, 92–96. https://doi.org/10.1109/ICISIP.2004.1287631

Borrmann, A., & Berkhahn, V. (2018). Principles of Geometric Modeling. In A. Borrmann, M. König, C. Koch, & J. Beetz (Eds.), Building Information Modeling: Technology Foundations and Industry Practice (pp. 27–41). Springer International Publishing. https://doi.org/10.1007/978-3-319-92862-3_2

Borrmann, A., König, M., Koch, C., & Beetz, J. (2018). Building Information Modeling Technology Foundations and Industry Practice: Technology Foundations and Industry Practice. In Building Information Modeling: Technology Foundations and Industry Practice. https://doi.org/10.1007/978-3-319-92862-3

Bosché, F., Ahmed, M., Turkan, Y., Haas, C. T., & Haas, R. (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. https://doi.org/10.1016/j.autcon.2014.05.014

Campagnolo, D., Camuffo, E., Michieli, U., Borin, P., Milani, S., & Giordano, A. (2023). Fully Automated Scan-to-BIM Via Point Cloud Instance Segmentation. Proceedings - International Conference on Image Processing, ICIP, 291–295. https://doi.org/10.1109/ICIP49359.2023.10222064

Cazorla, R., Poinel, L., Papadakis, P., & Buche, C. (2021). Bottleneck Identification to Semantic Segmentation of Industrial 3D Point Cloud Scene via Deep Learning. Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21), 4877–4878. https://doi.org/10.24963/ijcai.2021/670

Chen, S., Fang, J., Zhang, Q., Liu, W., & Wang, X. (2021). Hierarchical Aggregation for 3D Instance Segmentation. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 15447–15456. http://arxiv.org/abs/2108.02350

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 226–231.

Fröhlich, C., & Mettenleiter, M. (2004). Terrestrial laser scanning--new perspectives in 3D surveying. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36(8), W2.

Fumarola, M., & Poelman, R. (2011). Generating virtual environments of real world facilities: Discussing four different approaches. Automation in Construction, 20(3), 263–269. https://doi.org/10.1016/j.autcon.2010.08.004

Gao, B., Pan, Y., Li, C., Geng, S., & Zhao, H. (2021). Are We Hungry for 3D LiDAR Data for Semantic Segmentation? A Survey of Datasets and Methods. IEEE Transactions on Intelligent Transportation Systems, 1–19. https://doi.org/10.1109/tits.2021.3076844

Grilli, E., Menna, F., & Remondino, F. (2017). A review of point clouds segmentation and classification algorithms. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, 42(2W3), 339–344. https://doi.org/10.5194/isprs-archives-XLII-2-W3-339-2017

Han, L., Zheng, T., Xu, L., & Fang, L. (2020). Occuseg: Occupancy-aware 3d instance segmentation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2940–2949. https://doi.org/10.1109/CVPR42600.2020.00301

ISO. (2024). ISO 16739-1:2024: Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries – Part 1: Data schema (Issue ISO 16739-1:2024). ISO Geneva.

Justo, A., Lamas, D., Sánchez-Rodríguez, A., Soilán, M., & Riveiro, B. (2023). Generating IFC-compliant models and structural graphs of truss bridges from dense point clouds. Automation in Construction, 149, 104786. https://doi.org/10.1016/j.autcon.2023.104786

Kumar, N., Verma, R., Sharma, S., Bhargava, S., Vahadane, A., & Sethi, A. (2017). A Dataset and a Technique for Generalized Nuclear Segmentation for Computational Pathology. IEEE Transactions on Medical Imaging, 36(7), 1550–1560. https://doi.org/10.1109/TMI.2017.2677499

Lloyd, S. (1982). Least squares quantization in PCM. IEEE Transactions on Information Theory, 28(2), 129–137. https://doi.org/10.1109/TIT.1982.1056489

Ma, J. W., Czerniawski, T., & Leite, F. (2020). Semantic segmentation of point clouds of building interiors with deep learning: Augmenting training datasets with synthetic BIM-based point clouds. Automation in Construction, 113(March), 103144. https://doi.org/10.1016/j.autcon.2020.103144

Macher, H., Landes, T., & Grussenmeyer, P. (2017). From point clouds to building information models: 3D semi-automatic reconstruction of indoors of existing buildings. Applied Sciences, 7(10), 1–30. https://doi.org/10.3390/app7101030

Mafipour, M. S., Vilgertshofer, S., & Borrmann, A. (2023). Automated geometric digital twinning of bridges from segmented point clouds by parametric prototype models. Automation in Construction, 156. https://doi.org/10.1016/j.autcon.2023.105101

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

Mehranfar, M., Braun, A., & Borrmann, A. (2024). From dense point clouds to semantic digital models: End-to-end AI-based automation procedure for Manhattan-world structures. Automation in Construction, 162. https://doi.org/10.1016/j.autcon.2024.105392

Noichl, F., Collins, F. C., Braun, A., & Borrmann, A. (2024). Enhancing point cloud semantic segmentation in the data-scarce domain of industrial plants through synthetic data. Computer-Aided Civil and Infrastructure Engineering, 39, 1530–1549. https://doi.org/10.1111/mice.13153

Ochmann, S., Vock, R., Wessel, R., & Klein, R. (2016). Automatic reconstruction of parametric building models from indoor point clouds. Computers and Graphics (Pergamon), 54, 94–103. https://doi.org/10.1016/j.cag.2015.07.008

Pan, Y., Braun, A., Brilakis, I., & Borrmann, A. (2022). Enriching geometric digital twins of buildings with small objects by fusing laser scanning and AI-based image recognition. Automation in Construction, 140, 104375. https://doi.org/10.1016/j.autcon.2022.104375

Pan, Y., Noichl, F., Braun, A., Borrmann, A., & Brilakis, I. (2022). Automatic creation and enrichment of 3D models for pipe systems by co-registration of laser-scanned point clouds and photos. Proceedings of the 2022 European Conference on Computing in Construction, 3. https://doi.org/10.35490/EC3.2022.181

Patraucean, V., Armeni, I., Nahangi, M., Yeung, J., Brilakis, I., & Haas, C. (2015). State of research in automatic as-built modelling. 32nd International Symposium on Automation and Robotics in Construction and Mining: Connected to the Future, Proceedings, January 2015. https://doi.org/10.22260/isarc2015/0024

Rausch, C., & Haas, C. (2021). Automated shape and pose updating of building information model elements from 3D point clouds. Automation in Construction, 124(January), 103561. https://doi.org/10.1016/j.autcon.2021.103561

Schnabel, R., Wahl, R., & Klein, R. (2007). Efficient RANSAC for point-cloud shape detection. Computer Graphics Forum, 26(2), 214–226. https://doi.org/10.1111/j.1467-8659.2007.01016.x

Smith, A., & Sarlo, R. (2021). Automated extraction of structural beam lines and connections from point clouds of steel buildings. Computer-Aided Civil and Infrastructure Engineering, 1–16. https://doi.org/10.1111/mice.12699

Son, H., Kim, C., & Kim, C. (2015). 3D reconstruction of as-built industrial instrumentation models from laser-scan data and a 3D CAD database based on prior knowledge. Automation in Construction, 49, 193–200. https://doi.org/10.1016/j.autcon.2014.08.007

Son, H., Kim, C., & Turkan, Y. (2015). Scan-to-BIM-an overview of the current state of the art and a look ahead. 32nd International Symposium on Automation and Robotics in Construction and Mining: Connected to the Future, Proceedings, 1. https://doi.org/10.22260/isarc2015/0050

Souza, R. P. De, Santos, P. I. N., Franco, C. A. S., & Raposo, A. B. (2019). Automatic detection of 3d steel structures in as-built point clouds. ACM International Conference Proceeding Series, 147–154. https://doi.org/10.1145/3373509.3373565

Suchocki, C., Rutkiewicz, A., Katzer, J., & Kowalska, E. (2024). Remote estimation of cable tension using catenary theory and point clouds obtained by terrestrial laser scanning. Measurement: Journal of the International Measurement Confederation, 236. https://doi.org/10.1016/j.measurement.2024.115147

Szabo, A., Haaser, G., Steinlechner, H., Walch, A., Maierhofer, S., Ortner, T., & Gröller, E. (2023). Feature-assisted interactive geometry reconstruction in 3D point clouds using incremental region growing. https://doi.org/10.1016/j.cag.2023.02.004

Tagliasacchi, A., Delame, T., Spagnuolo, M., Amenta, N., Telea, A. C., & Telea, A. (2016). 3D Skeletons: A State-of-the-Art Report. Computer Graphics Forum, 35(2), 573–597. https://doi.org/10.1111/cgf.12865

Volk, R., Stengel, J., & Schultmann, F. (2014). Building Information Modeling (BIM) for existing buildings - Literature review and future needs. Automation in Construction, 38(October 2017), 109–127. https://doi.org/10.1016/j.autcon.2013.10.023

Wang, B., Yin, C., Luo, H., Cheng, J. C. P., & Wang, Q. (2021). Fully automated generation of parametric BIM for MEP scenes based on terrestrial laser scanning data. Automation in Construction, 125, 103615. https://doi.org/10.1016/j.autcon.2021.103615

Wang, Q., Guo, J., & Kim, M.-K. M. K. (2019). An application oriented scan-to-bim framework. Remote Sensing, 11(3). https://doi.org/10.3390/rs11030365

Wang, Q., Tan, Y., & Mei, Z. (2019). Computational Methods of Acquisition and Processing of 3D Point Cloud Data for Construction Applications. Archives of Computational Methods in Engineering, 0123456789. https://doi.org/10.1007/s11831-019-09320-4

Won Ma, J., Jung, J., & Leite, F. (2024). Deep Learning–Based Scan-to-BIM Automation and Object Scope Expansion Using a Low-Cost 3D Scan Data. Journal of Computing in Civil Engineering, 38(6). https://doi.org/10.1061/JCCEE5.CPENG-5751

Xie, Y., Tian, J., & Zhu, X. X. (2020). Linking Points with Labels in 3D: A Review of Point Cloud Semantic Segmentation. IEEE Geoscience and Remote Sensing Magazine, 8(4), 38–59. https://doi.org/10.1109/MGRS.2019.2937630

Yan, Y., & Hajjar, J. F. (2021). Automated extraction of structural elements in steel girder bridges from laser point clouds. Automation in Construction, 125, 103582. https://doi.org/10.1016/j.autcon.2021.103582

Yan, Y., & Hajjar, J. F. (2022). Geometric models from laser scanning data for superstructure components of steel girder bridges. Automation in Construction, 142. https://doi.org/10.1016/j.autcon.2022.104484

Yang, L., Cheng, J. C. P., & Wang, Q. (2020). Semi-automated generation of parametric BIM for steel structures based on terrestrial laser scanning data. Automation in Construction, 112(112), 103037. https://doi.org/10.1016/j.autcon.2019.103037

Yin, C., Wang, B., Gan, V. J. L., Wang, M., & Cheng, J. C. P. (2021). Automated semantic segmentation of industrial point clouds using ResPointNet++. Automation in Construction, 130(July), 103874. https://doi.org/10.1016/j.autcon.2021.103874