



# Scan2FEM: From Point Clouds to Structured 3D Models Suitable for Simulation

Z. Selman<sup>1</sup> , J. Musto<sup>1</sup>, and L. Kobbelt<sup>1</sup> 

<sup>1</sup>RWTH Aachen University, Germany

## Abstract

*Preservation of cultural heritage is important to prevent singular objects or sites of cultural importance to decay. One aspect of preservation is the creation of a digital twin. In case of a catastrophic event, this twin can be used to support repairs or reconstruction, in order to stay faithful to the original object or site. Certain activities in prolongation of such an objects lifetime may involve adding or replacing structural support elements to prevent a collapse. We propose an automatic method that is capable of transforming a point cloud into a geometric representation that is suitable for structural analysis. We robustly find cuboids and their connections in a point cloud to approximate the wooden beam structure contained inside. We export the necessary information to perform structural analysis, on the example of the timber attic of the UNESCO World Heritage Aachen Cathedral. We provide evaluation of the resulting cuboids' quality and show how a user can interactively refine the cuboids in order to improve the approximated model, and consequently the simulation results.*

## CCS Concepts

• **Computing methodologies** → **Point-based models**; • **Applied computing** → **Engineering**; **Digital libraries and archives**;

## 1. Introduction

Decay of historic objects is an ever-present challenge that is relevant for scientists from a variety of backgrounds. Preservation of culturally relevant objects and sites is commonly supported by digitizing them, in order to obtain a digital twin that is not affected by decay over time. Additionally, a digital object enables presentation to a large audience without having to provide physical access to the protected site or object. Furthermore, a digital twin can be utilized for inspection of structural integrity. Defects that could lead to catastrophic events, such as structural collapse, can be identified and averted in advance by *e.g.* addition of supporting elements. Conversely, an existing digital twin can guide the restoration process after a catastrophic event.

Often the sites and objects to be digitized existed for centuries and have to be treated with care to avoid further decay, requiring non-intrusive acquisition techniques. Various technologies can be utilized to create digital twins in the form of point clouds, such as photogrammetry and 3D laser scanning. Both technologies allow non-intrusive digitization, and the ability to capture large structures. Advantages of using laser scanners include higher geometric precision, the ability to capture at greater distances and being invariant to low light conditions.

Material decay over time reduces structural integrity and it may be necessary to add supporting structures. An expert might have an intuition on how to achieve this, however an evidence-based approach requires measurements and simulation. Various ro-

bust pipelines exist to visualize, or perform measurements in scan data [Rem03; Gir16]. For computational simulation, a tool to obtain a representation suitable for structural analysis of historic structures does not exist. Structural simulations require professionals that hand-craft the model which is used for structural analysis of rigid bodies, where point clouds and photographs are only used as a visual reference but rarely as the raw input data that is converted directly into a simulation-suitable representation.

Solving the general problem of automatic structural analysis from scan data is not straightforward. A way to reduce the problem to a solvable subproblem is by restricting the set of possible objects to a subset of all real-world objects. When considering structural integrity of historic buildings, one reasonable subset are its supporting structures, especially in the attic, as those are mostly accessible and not surrounded by unrelated objects. As these supporting structures are often made from wooden beams with rectangular cross-sections, our method aims to solve the restricted problem for timber structures with cuboidal beams.

We propose an automated method that computes beam geometry and connectivity directly from point clouds, that can be used in simulation software for structural analysis. Refinement is necessary in situations where only limited access to certain areas is possible, or when cuboids are incorrectly reconstructed. Thus, it is additionally possible to further refine the obtained model prior to structural analysis.

We make the following distinction in this paper: a *beam* refers

to the physical timber object located *e.g.* in the attic of a historic building. A *cuboid* is the digital twin of a beam defined by eight connected corner points.

In summary, we make the following contributions:

- An automated pipeline to obtain a cuboid-based representation of a point cloud, suitable for processing with simulation software
- An interactive mode to refine the representation prior to simulation
- An improved RANSAC-based approach to detect planar regions
- An evaluation of our method based on the timber attic of the Aachen Cathedral

The remainder of this paper is structured as follows: Section 2 discusses previous approaches to computing suitable simulation representations from scan data. In Section 3 we provide a detailed explanation of our method leading to a short outline of details regarding our data and acquisition method in Section 4.1. We present results on the steps of our method in Section 4 and discuss the quality of the results and some limitations of the method.

## 2. Related Work

[THA\*10; VSS14] provide a comprehensive analysis of challenges and approaches in automatic reconstruction of building information. In the following, we offer insight into several approaches relevant to our work.

Research on preservation of cultural heritage through digitization has existed for over two decades. One prominent example is the digital Michelangelo project [LPC\*00], where Levoy *et al.* digitized Michelangelo's David sculpture. Furthermore, various projects digitize large scale physical objects using different acquisition methods, such as Structured-light 3D scanners [RCM\*01] and Structure from Motion [HZ03].

Research concerning robust and efficient estimation of normal information in unstructured 3D data include [MWP18; MKT21]. Mura *et al.* [MWP18] iteratively refine a normal vector candidate based on a discrete neighborhood. Their method estimates robust normal vectors, significantly faster compared to methods based on sampling consensus. In [MKT21] Molnar *et al.* leverage a learning-based approach, to estimate normals for depth images from Time-of-Flight cameras. The network allows for robust estimation suitable for real time applications. As normal information is one of the inputs to our method, especially the plane-fitting stage benefits from accurate and feature-preserving normals.

Work on FEM simulation of structures has been done by [ALOR09; CDB\*15]. In [ALOR09] Armesto *et al.* obtained their data from digitizing photographs of a timber truss structure. After manually reconstructing one girder, they evaluate a variety of stresses on it. [CDB\*15] propose a semi-automated pipeline with several manual steps throughout. Eventually, a voxel model is obtained from the initial point cloud which is structurally analyzed. Our focus lies on the analysis of a complete structure, as close as possible to the original representation especially for arbitrarily oriented beams.

Various methods exist, to reconstruct a part of a building digitally. Yang *et al.* [YKG17] reconstructed the truss structure of

Château du Haut-Koenigsbourg. Making use of a total station, for each beam they obtained at least six points on parallel edges. Leveraging the association between beams and points, the sparse information is sufficient to reconstruct the beams. However, this prior knowledge is not available for general point clouds. Notable works focusing on automatic reconstruction from point data include [XTHS18; WXW18; XAAH13; VAC12]. In [XTHS18] the authors use a novel 3D local feature descriptor to reconstruct scaffolds from scan data of construction sites. Their descriptor is based on a feature histogram which discretizes normal orientation, which is not required in our approach. In order to separate points between different types of objects in the scaffold, they perform a supervised random forests classification. A method to construct 3D CSG models from point clouds is presented by [WXW18]. By identifying surface patches and reconstructing geometric primitives to fit them, the authors obtain a set of candidates whose union creates a CSG model of the object. Similar to [XAAH13] they rely on voxelization of regions, which results in a trade-off between resolution and memory/time constraints. [VAC12] propose a reconstruction technique based on RFID tags in order to identify a certain type of object. Depending on the type, points belonging to this object are removed. In order to reconstruct remaining objects, *e.g.* walls, their data is discretized as well. Pöchtrager *et al.* assume a certain range of valid beam sizes which is not the case for general point clouds. An alternative solution to sub-segment splitting is provided Murtiyoso *et al.* [MG20], which is constrained to Y/L-shaped segments. However, as highlighted in Section 4.5 we manage to compute this split with the bare minimum of assumptions, which allows even complex regions to be split. Özkan *et al.* [ÖPS\*22] propose a similar approach to our method. Despite many similarities, the authors require a technique for removing roof cover from the scan data. As this step may erroneously remove important data, our method avoids performing any filtering on the scan data prior to plane fitting. A detailed discussion of the shortcomings of these methods is provided in Section 4.2.

## 3. Method

Our method performs a bottom-up feature analysis starting with simple local planarity analysis (Section 3.2). These planar regions (PRs) are consolidated to six-sided cuboids representing individual beams (Section 3.3). They are then integrated into the global structure by computing a connectivity graph based on cuboid intersections (Section 3.4).

### 3.1. Parameters

When dealing with unstructured and noisy data, we require several parameters and thresholds at different stages of the pipeline. Some of them have an interpretation that follows directly from the data itself, *e.g.* aspect ratio (AR) of what is detected as a beam. Some, like minimal number of points in a valid region, depend on sampling density and filtering. They provide a stopping criterion for certain procedures to avoid degenerate scenarios. Values, for parameters introduced in this chapter, are listed in Table 1.

### 3.2. Planar Region Processing

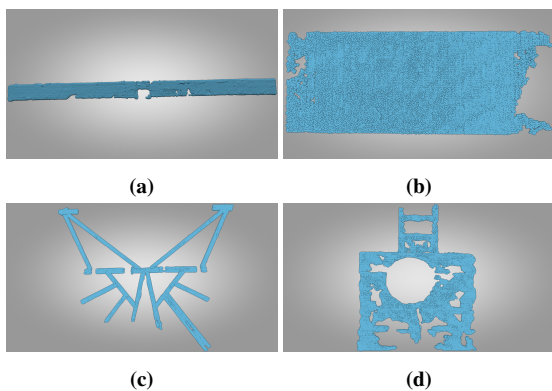
Geometric scan data, 3D points  $p_i \in \mathbb{R}^3$  and per-point oriented normal information  $n_i \in \mathbb{R}^3$ , serves as input to our method. Using a RANSAC-based approach for point cloud segmentation proposed by Poux *et al.* [PMSK22], we identify PRs as clusters of points that are close to a plane, and whose normals point into a similar direction as the plane normal. Each PR embodies one side of a scanned beam's surface. However, not all planar surfaces are beams. Some PRs may originate from different objects such as walls, floors, or doors. We use a classification based on AR and density to identify whether a PR originated from a beam. Also, although an obtained region is planar, it may consist of union of multiples regions from different beams, *e.g.* when two or more level beams intersect. An example for this is visualized in Fig. 1c. Thus, prior to cuboid fitting it may be necessary to split them, as explained in Section 3.2.1.

Our core assumption about well-formed beams is that they are sufficiently sampled and significantly longer than wide. Consequently, this property holds for PRs as well. In order to distinguish well-formed from malformed PRs, we classify them into three categories, based on AR and density:

1. dense and  $AR \geq \theta \Rightarrow$  valid (saved)
2. dense and  $AR < \theta \Rightarrow$  invalid (discarded)
3. not dense  $\Rightarrow$  contains multiple PR (further processed)

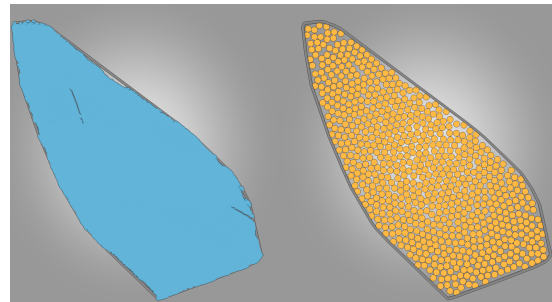
where  $\theta$  depends on the specific data set being processed. Due to the simple geometrical interpretation, *i.e.*, ratio of length by width, it is not difficult to estimate  $\theta$  (Table 1). An example PR from our dataset for each class is shown in Fig. 1.

Each PR can easily be transformed into a plane. Reducing the data dimensionality to 2D simplifies several steps of our method, *e.g.* calculation of density, AR, or intersections. For this, we estimate the orientation of each PR using principal component analysis (PCA) on the covariance matrix of the PRs 3D points. Here, the *principal direction* refers to the eigenvector with the largest eigenvalue. The *normal* refers to the eigenvector with the smallest eigenvalue. The remaining eigenvector is referred to as *tangent*.



**Figure 1:** PRs visualize each class. Fig. 1a and Fig. 1b show classes 1 and 2. Both are quite dense but Fig. 1b lacks sufficient AR. Both images on the bottom row are class 3 where Fig. 1c is a desirable PR, as meaningful PRs can be extracted, and Fig. 1d is an undesirable one.

The AR of a PR is calculated using a minimal oriented bounding box based on the principal direction and tangent. In order to estimate if a PR is dense, we use the convex hull to obtain a closed contour. It approximates the PR boundary, given it is a well-formed beam side. At each 3D sample a disk of a radius  $\tau$  (Table 1) is placed. Next, we gather all other points closer than  $2\tau$  to it and remove them. This is repeated for the next disk. Eventually, we obtain a set of disks covering the contour without overlap, representing the coverage by the input points as visualized in Fig. 2. Another example is located in the supplementary material. A PR is dense, if the ratio of the area of sum of disks and the area of the contour is close to 1. If the ratio is significantly smaller, it indicates existence of holes or a union of PRs that require splitting.



**Figure 2:** Example of density computation. We replace each point by a disk, and remove all other points in the neighborhood of that disk to reduce overlap. This is repeated until all points are either replaced by a disk or removed. The result is a non-overlapping sampling of disks inside the convex-hull, which can be used to estimate density. In this example the density is  $\sim 88\%$ .

#### 3.2.1. Planar Region Segmentation

All PRs that fall into category 3 are processed further to identify potentially new PRs. An example for such a PR can be observed in Fig. 1c. If the PR is a union of multiple PRs, it is possible to find multiple center-lines that represent the principal direction of each new region. Finally, one has to estimate the width of the new PR with respect to the center-line and collect all points belonging to it. In the following, we propose a method to estimate these center-lines using Hough transformation.

Based on the prior assumptions about PRs, long sub-region are the most likely candidate for a new PR. We can identify long sub-regions by exercising the following probabilistic approach: we start and select random points from the PR and trace multiple rays in uniformly distributed random directions. Next, we intersect each ray with the 2D  $\alpha$ -shape [EM94] (using the implementation provided by CGAL [The22]) and calculate the segment lengths. As we trace from the inside and as the contour is closed, we are guaranteed at least two intersections that form a segment. Each line is Hough transformed [DH72] and weighted by its length, which emphasizes long sub-regions. After all lines are transformed, we identify the maximum in Hough space and transform it back to obtain a center-line.

A possible reconstruction method for the center-line could be to identify the mode with Mean shift [FH75]. In practice, the accuracy

is comparable to using the maximum. Thus, we use the newly obtained line and points as a robust initialization for a small number of iterations of least squares line fitting. As the initial line and set of points are a good estimate, we observe reliable convergence.

The back-transformed line corresponds to the principal direction of the new PR, however it is still necessary to identify which points belong to it. A reliable way to approximate the width is by placing multiple equidistant samples on the center-line. From these points we cast a ray perpendicularly to the principal direction, compute the ray-contour intersections and measure their lengths. We compute the mode of this length distribution and obtain a robust width estimate  $\omega$  of the new PR. All points closer than  $\frac{\omega}{2}$  to the line belong to the new PR. Additionally, these points are removed, while remembering the set of original points. Removing them reduces the chance that the same region is extracted more than once when generating a new set of sample lines. It also provides a termination condition, namely when a high percentage of points has been removed. Selection from the full set of points enables new PRs to overlap. This avoids later regions to have low density and improves the connectivity of reconstructed cuboids. Here, a PR is rejected, if it falls into any other category than 1 (cf. Section 3.2) or if its number of points is less than  $p_{min}$ . Each new PR is classified again, with relaxed conditions as valid but shorter PR might emerge from this procedure. The user may specify the maximum attempts  $L$  (Table 1) to extract a new PR.

### 3.3. Cuboid Generation

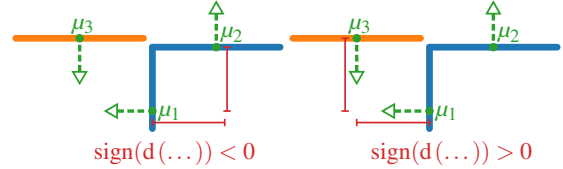
After processing each PR, we search for all pairs of valid PR that could make up a cuboid representing a beam, based on three conditions: Orthogonality, Normal Orientation, Distance. Wooden beams from an attic may have arbitrary orientation in space, with varying lengths and widths. One similarity between all of them is their rectangular cross-section. We leverage this assumption for grouping PRs to form a cuboid.

**Orthogonality** A PR is considered side of a beam, if we can find another PR that is orthogonal to it and points into the same principal direction. Given the prior orientation estimation, we can calculate this condition in the following way: two PRs are orthogonal to each other, if the angle between the normal vectors is  $90^\circ$ . Both PRs point into the same direction, if the angle between the principal directions is  $0^\circ$  or  $180^\circ$ . In practice, angles are never perfectly aligned. For all angle comparisons of normal vectors and principal directions, we allow a few degrees deviation.

**Normal Orientation** Two PRs that are orthogonal to each other, might not necessarily belong to the same beam. Suppose the cross-section of three PRs where one is perpendicular to the other two, as shown in Fig. 3. Here, both blue PRs are supposed to form a pair, while the orange PR belongs to another one. We tell both situations apart by checking on which side of one PR the other is. As all pairs are orthogonal to each other, one PR has to lie *in front* of the average point of the other PR in terms of its normal vector. We verify this by checking if the PR is *behind* the side we are currently inspecting, *i.e.*, has negative distance.

Leveraging the oriented normal information of the scan data, we construct two planes ( $\rho_1, \rho_2$ ) and calculate the average points

( $\mu_1, \mu_2$ ) as representatives of two PR. It suffices to check, if the point-plane distances have consistent signs with themselves and the normal orientation *i.e.*,  $\text{sign}(d(\mu_1, \rho_2)) = \text{sign}(d(\mu_2, \rho_1))$ . If the normals point to the outside, *i.e.*, towards the scanner, the sign is negative, positive otherwise. If the signs are not equal or not consistent with the convention of normal orientation, the PRs do not form a pair.



**Figure 3:** Here, a cross-section of three PRs which may form a pair is shown. Both normals at  $\mu_2$  and  $\mu_3$  are perpendicular to  $\mu_1$ . In both cases the pairwise distance has consistent sign, however the normal vector (green) points to the outside. Thus both blue PRs have consistent sign with the normal (negative), while on the right both distances have positive sign.

**Distance** Ultimately, both PRs have to be close to each other. We check whether the pair of closest points from both regions are less than a few centimeters apart.

If all of the prior conditions are satisfied both PRs form a pair. After gathering all pairs, transitive pair connections are established and grouped. Suppose each pair is given as tuple of indices ( $i, j$ ) with  $i \neq j$ , we group all tuples that transitively share one index. *E.g.* the given pairs (0, 1), (1, 15), (0, 20), (2, 3), and (3, 42) result in two groups: (0, 1, 15, 20) and (2, 3, 42). Grouping enables a split PR to belong to the same cuboid again, if they share a common PR in their tuples.

From each group we construct a cuboid with the goal to closely represent its wooden beam, by generating a cross-section from all points in the group and extruding it to fit all of them. First, collect all points of the group and calculate the mean principal direction  $\delta$  and point  $\mu$ . Then, project all points of the group into a plane  $\rho$  with  $\rho = (\delta, \mu)$  to obtain a set of points representing the beam's cross-section in 2D. In order to make the estimation of the cross-section more robust, we remove outliers of each PR in the group. This is achieved by projecting the group's tangential vectors into the plane as well. By additionally projecting all points onto the transformed tangential vectors, we can remove a small percentage of top/bottom points in both directions. Next, we calculate the convex hull and simplify it, until we obtain a quadrangular polygon. This is achieved by iteratively removing those vertices, that introduce the smallest change in the area of the polygon, until only four vertices are left. Some groups may only have two PRs, causing the cross-section's hull from this procedure to degenerate into a triangle. In this case, we mirror the points before computing the hull to avoid this from happening.

The four remaining vertices have angles close to  $90^\circ$  and form the cross-section of the new cuboid. In a final step, we estimate the group's length along  $\delta$  by projecting all points onto it and selecting the minimum and maximum value. Finally, we obtain a six-sided

cuboid, by placing two copies of the cross-section at each end of the group and connecting them. This is repeated for all groups and results in a soup of cuboids.

### 3.4. Cuboid Connectivity

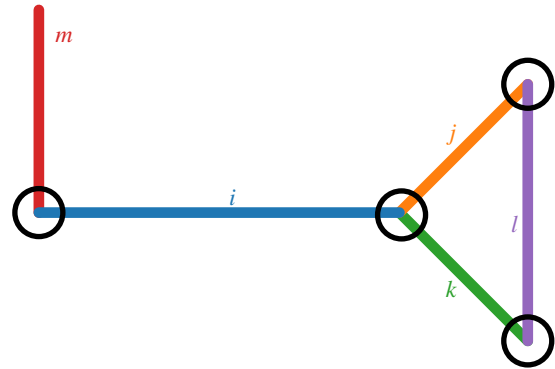
After obtaining a soup of cuboids, a multitude of them intersects and their connectivity can be modeled as a graph. We create the graph by constructing the sets of cuboid that produce a valid intersection and connecting their intersections. This graph serves as basis for the structural analysis. We notice that some cuboids exist which *should* intersect but do not, as can be observed in Fig. 12. This is problematic for structural analysis, as missing connectivity in places where there should be a connection leads to drastically different simulation results. One of the most common reasons for the lack of connectivity is occlusion when acquiring data. Close to joints where multiple beams meet, each beam contributes to the overall coverage reducing the number of samples in the resulting point cloud at the respective location. Our method offers a way for users to interactively extend cuboids, such that they intersect in the desired location. Already existing intersections are modeled using an underlying graph, where each edge represents a cuboid and each vertex represents an intersection of two or more cuboids. Introducing new intersections by user interaction, in turn updates the underlying graph as visualized in Fig. 5.

#### 3.4.1. Graph Creation

The automatically detected intersections of cuboids serve as basis for the initial graph structure which is later interactively updated by the user. One requirement for the graph structure is, that in places where multiple cuboids intersect each other, only one node is generated. Otherwise, a joint where three or more cuboids intersect, would consist of multiple joints. This makes the straightforward approach using only pairwise intersections inadequate.

A visualization of the following scenario is given in Fig. 4. First, we compute pair-wise intersection tests to obtain information about which cuboids intersect with each other, e.g.  $N = \{i: \{j, k, m\}, j: \{i, k, l\}, k: \{i, j, l\}, l: \{j, k\}, m: \{i\}\}$  with  $i, \dots, m \in C$ , where  $C$  is the set of all cuboids. Here,  $i: \{j, k, m\}$  denotes that  $i$  (referred to as key) intersects with  $j, k$ , and  $m$  (referred to as associated set). Next, the goal is to obtain a set  $D$  of all sets  $\hat{D}$  that form non-empty intersections from each element of  $C$ . From each entry  $n \in N$ , we construct a set  $\hat{D}$  that contains a decomposition of each  $n$ , such that each  $\hat{D}$  yields non-empty intersections, in the following way: Each entry  $n \in N$  is decomposed by creating a set  $\hat{D}$  from the key and the first element of the associated set, as they are guaranteed to have an intersection. For each remaining element of the associated set we test whether an intersection with  $\hat{D}$  results is empty. If yes, it is added to  $\hat{D}$ , and  $\hat{D}$  is added to  $D$ . If not, we continue with the next element of the associated set. Eventually, the combination of key and first element of associated set is exhausted, and we repeat this process with the key and the second of the associated set. Duplicate partitions and pure subsets (i.e.,  $\{j, k\}$ ) are eliminated from  $D$ . This results in:  $D = \{\{i, j, k\}, \{i, m\}, \{j, l\}, \{k, l\}\}$ , because  $m$  does not intersect with  $j$  or  $k$ , and intersecting  $\{j, k, l\} = \emptyset$  as shown

in Fig. 4. Each  $\hat{D} \in D$  corresponds to the largest set of cuboids that yields a non-empty intersection.



**Figure 4:** Visualization of how intersections of cuboids  $i, j, k, l, m$  are represented in the graph. Despite  $j, k, l$  intersecting with each other, we have to construct partitions as their joint intersection is empty. Each black circles visualizes what a calculated partition captures: the largest set of cuboids that yield a non-empty intersection.

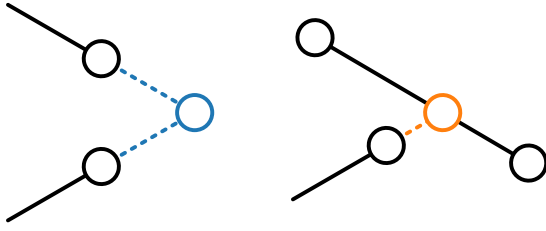
For each partition we calculate the intersection of all involved cuboids. We pick a representative point in the intersection volume and consequently inside all cuboids in the partition. Each of these points serve as vertex of the underlying graph structure. For every cuboid we collect all points inside it, sort them along the principal direction of this cuboids and connect them. Cuboids that do not intersect any other cuboid are represented as isolated edge with two vertices on both ends.

#### 3.4.2. Interactive Editing

In order for the user to quickly refine the cuboids, we provide the possibility for a user to interactively perform certain operations on the cuboids where deemed necessary.

One interaction is the extrusion of cuboids along their principal axis if they intersect with another cuboid afterwards. We consider this interaction to be the most important one, as missing intersections between cuboids are noticeable in the structural analysis. A user may select pairs of cuboids to intersect them by clicking on them. By extruding a copy of those cuboids in both directions by a chosen amount, the user can interactively select which cuboids should intersect and at which distance. During the interaction there is a marker indicating where the intersection occurs, and a confirmative action extrudes both cuboids respectively. A graph update is performed by checking whether both cuboids intersect outside of each other if one cuboid reaches into the other. In the first case both graph vertices are moved to the target intersection location, otherwise the edge corresponding to the respective cuboid is split at the intersection location, as shown in Fig. 5.

Some cuboids in the collection may be incorrect or undesirable, e.g. correct beams that do not contribute meaningfully to further steps. Despite best efforts, the automated planar segmentation may generate an incorrect or badly oriented new PR. Thus, it is also possible to simply remove undesirable cuboids by selecting them and confirming their removal.



**Figure 5:** Visualization of the two possible graph updates for cuboid intersections. If the new position is outside of both cuboids (left), we move both vertices towards the target position and merge them. If it is contained in one cuboid (right), only one vertex is moved and the other edge is split at the target position.

## 4. Results and Evaluation

In this section the results obtained in various steps of our method are presented. First, we introduce the dataset that we evaluated our method on (Section 4.1). Next, we briefly discuss the differences to methods presented in Section 2, followed by an overview of the planar fitting (Section 4.3). Next, we evaluate the planar segmentation and discuss improvements and limitations (Section 4.4). Lastly, an overview over the cuboid reconstruction, refinement by user interactions (Section 4.5) followed by a structural analysis of the reconstruction is given (Section 4.6).

### 4.1. Evaluation Dataset

We apply our method on data acquired in the UNESCO World Heritage Aachen Cathedral in Germany. By applying the method to the point cloud of the attic, we can reconstruct wooden beams responsible for stability and perform structural analysis. The data largely consists of a collection of timber beams that sustain the whole attic. We used a Riegl VZ-400 Terrestrial Laser Scanner and a Leica BLK360 Imaging Laser Scanner. Consisting of 36 scan positions and roughly 54 million points, after grid sampling and outlier detection, the data set is  $\sim 33$  m long,  $\sim 14$  m wide, and  $\sim 20$  m high. Each point  $p_i \in \mathbb{R}^3$  in this point cloud is assigned a normal vector  $n_i \in \mathbb{R}^3$ , which is oriented towards the outside of the surface, *i.e.*, towards the scanner. As long as not specified otherwise, all references to normal information will imply this orientation.

In Table 1 we specified the parameters used for the evaluation of our method on the Dataset of the Aachen Cathedral

**Table 1:** Parameters of our method used for this dataset.

Parameter	$\tau$	$\theta$	$p_{min}$	$L$
Value	2.5 cm	4.0	300	32

### 4.2. Differences to Similar Methods

A comparison with other contributions in the area of scan data processing mentioned in Section 2 is difficult. As neither their data set nor their code is published a direct evaluation is not possible. However, certain parts of their algorithm exhibit weaknesses, that we circumvent.

In their method Pöchtrager *et al.* [PSDP17; PSDP18] estimate *compactness*, which is the ratio between areas of  $\alpha$ -shape and minimum oriented bounding box, and ignore them for further processing. Our method proposes to compute a *density*, based on splatting in the convex hull, explained in Section 3.2. A low density indicates a potential union of PRs that we split to avoid the loss of potential PRs for cuboid generation.

A solution to sub-segment splitting is provided Murtiyoso *et al.* [MG20]. As opposed to their method, we manage to compute this split with the bare minimum of assumptions about the beam shape. This allows even complex regions to be split, as highlighted in Section 4.5. Additionally, they perform their method on octree bins instead of the points directly, which reduces the accuracy of the reconstruction and makes it less suitable for structural analysis. Despite their octree representation, they did not perform any experiments on datasets larger than  $\sim 100000$  points, whereas our dataset consists of over 50 million points.

Özkan *et al.* [ÖPS\*22] require the usage of a roof cover removal which may introduce unintended errors into the scan data. As noise in the data and potentially arbitrary shapes are possible, each filtering step in scan data introduces the possibility for errors. On the contrary our method leverages the oriented normal information to avoid obtaining cuboids near the roof cover and all other areas where no cuboid is present, as explained in Section 3.3. Additionally, our method allows to obtain a representation of scan data which is ready for structural analysis, in an automatic fashion. Note, that the quality of the model, and simulation time and accuracy usually improves with user interaction.

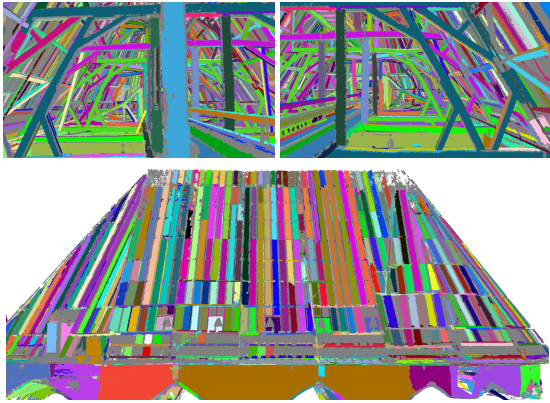
### 4.3. Plane Fitting

Starting with 54 million points, we obtain 8329 PRs from 47 million (87%) points. After discarding/gaining regions through the classification and planar segmentation, we obtain a total of 9117 PRs. Note, that due to the probabilistic nature of the planar segmentation, resulting PRs slightly vary.

Fig. 6 provides an inside and outside view of the scan data with colored PRs. Visible from the outside view (bottom row) one notices numerous planar regions located on the roof. Essentially all of these PRs are irrelevant as they do not contribute to the formation of any cuboid later on. The inside view (top row) visualizes the coverage of almost all relevant beams with PRs. Similar to the roof, there are some PRs that do not originate from a beam, which does not pose a problem as these regions do not find suitably oriented PRs to form a pair (cf. Section 3.3). In the rare cases that they do, *e.g.* cuboids forming based on a ladder lying on the floor, a final decision can be made by the user to discard these cuboids.

### 4.4. Segmentation of Planar Regions

An essential step for obtaining a more complete reconstruction, is the segmentation of planar regions. Large PRs that are not sufficiently split, reduce the amount of evidence for multiple beams at once. As each beam requires at least two PRs in order to be reconstructed, splitting up multiple PRs is essential to the overall completeness. Fig. 7a and Fig. 7b illustrate the split for some of the



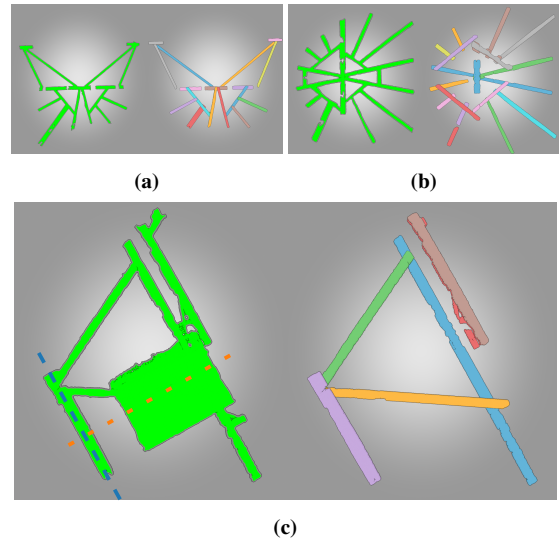
**Figure 6:** Splat rendering of the scan data with the plane fitting results. Different colors correspond to different planar regions. All gray splats are not part of a region. As indicated by the numbers and the coloration, the plane fitting finds the majority of relevant planar regions.

more complicated cases. In the top row, there are two examples for unions of a larger number of PRs in arbitrary orientation and connectivity. Additionally, several cavities that have been carved by the  $\alpha$ -shapes introduce additional difficulties. We have observed that the overall split is quite robust, despite some regions are missing or shorter than intended.

It is possible that the new line passes through the contour of the PR more than just twice. Particularly in situations where areas of the beam were not sampled sufficiently from the scanner, or contour generation using  $\alpha$ -shapes altered or separated these areas entirely. A center-line like this would simply collect sample points across boundaries, which may consume a whole valid region. This can be observed in Fig. 7c, where the backside of a wooden box, which is coplanar with several beams, produces the large green area with the orange dashed line. Its center-line is obtained before the blue line and has a very large width, which would consume the future violet region if extended beyond the contour. In order to prevent the future violet region to be contained inside it, we split the orange center-line at the contour. Each sub-segment that is inside individually collects points within the width and re-estimates its orientation. Here, this will cause no change to the initial large sub-region, as it will be discarded based on its AR. We re-estimate the center-line of the second sub-region, depicted in blue. It becomes a well-defined new PR colored violet.

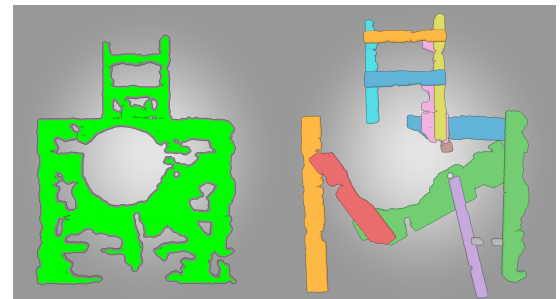
Fig. 7c also demonstrates the capabilities of PRs retrieval in regions where the majority of points does not stem from a valid cuboid. Here, the future orange PR proceeding diagonally across is largely occluded by the box's backside. Despite this, width estimation using the mode of the distribution of measured distances provides us with a realistic new PR.

Occasionally the planar segmentation is not as successful as shown in Fig. 8. Here, the region is no union of PRs, however our method can not detect the difference. It manages to extract seemingly plausible regions, which do not affect the quality of the cuboid fit too much. As each region has to fulfill certain criteria as



**Figure 7:** Results of the planar segmentation. On the left in green is the original planar region, on the right is the resulting split. Different colors represent different new regions. As our method does not assume any underlying model, even very complex constellations (Fig. 7a, Fig. 7b) or beams passing through very large regions (Fig. 7c) are separated robustly.

explained in Section 3.3, these regions are unlikely to find a matching region conforming to these criteria. Even in the case of an erroneous cuboid creation, the interactive editing enables their removal.



**Figure 8:** Example for a region that should not be split, however distinguishing between valid unions of regions and this case is not trivial. Multiple plausible PR are found, however they rarely form a pair.

A quantitative analysis of the planar segmentation is hard, as it would require a manually labeled ground truth data set. It is possible gain insight, by performing experiments limiting segmentation to observe change. Furthermore, it is possible to inspect the quality of cuboid reconstruction.

Table 2 provides insight on the benefits of planar segmentation. In this experiment, we limit the maximum number of segmentation attempts per PR. We observed an increase in number of extracted PRs and consequently cuboids, with increased limits. Increased number of used and total PRs suggest that some new PRs

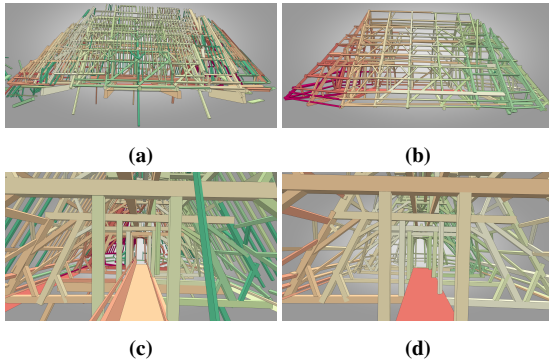
are undesired, as demonstrated in Fig. 8. Note, that the application is not utilizing full parallelization, which influences the reported runtime.

**Table 2:** Results for the experiment that limits the number of PR extractions and performs the remaining pipeline. We observe an increase in number of found cuboids, used and total PRs for each increase in the limit. Note that due to the probabilistic nature of the method, results may slightly vary between runs.

Limit	# Cuboids	Regions Used / Total	Duration
1	1081	3376 / 6838	44 min
2	1211	3837 / 7670	43 min
4	1317	4258 / 8337	49 min
8	1366	4571 / 8788	56 min
16	1393	4775 / 9029	56 min
32	1399	4847 / 9117	56 min

#### 4.5. Cuboid Reconstruction

In the final stage of the pipeline all remaining PRs are checked if they can form a cuboid. First, we give a brief visual overview over the results, followed by an analysis of the quality of the cuboid construction. In Fig. 9 we visualize the reconstruction results and optional steps that the user can take interactively, *e.g.* cuboid extrusion or removal. Visual inspection indicates that a high number of beams on the inside and outside are reconstructed as cuboids. Additionally, on the right some small beams were removed, due to their negligible contribution to structural integrity.



**Figure 9:** Fig. 9a shows the beams that have been obtained by our method, with several extrusion and removal operations performed in Fig. 9b. The second row shows the same respective data from an inside perspective. Reasons for removing cuboids include: faulty reconstructions in previous stages, or lack of contribution to the structural integrity of the object, *i.e.*, most of the thin cuboids on the roof sides. Their removal is optional, but impacts simulation time.

One way to assess the cuboid fitting quality is to measure the unsigned distance of all points to the cuboid they belong to. We plot the results of this measurement in Fig. 11. As the set of points belonging to a cuboid is known at the time of construction, we simply take the minimum distance of each point to all six cuboid sides.



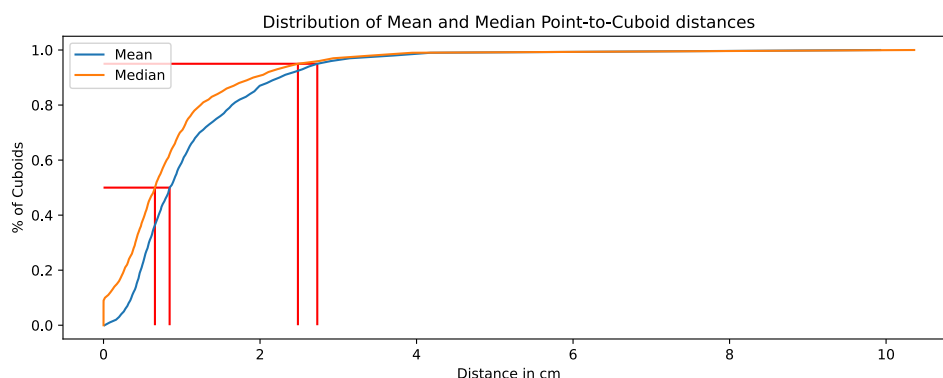
**Figure 10:** Here, different perspectives from inside the point cloud of points are visualized, colored according to distance to their cuboids. Pure violet corresponds to zero and pure yellow to the maximum distance of a point to its cuboid. Fig. 10a shows a close-up of several outliers. An example for a bad fit can be seen in Fig. 10b. Incorrect planar segmentation causes the cuboid to be larger than necessary.

By computing the mean and median distance over each beam, we can quantify the proportion of cuboids with a mean/median distance below a certain threshold. As can be discerned in Fig. 11, 50% of all cuboids have a mean/median distance well below 1 cm, 95% are well below 3 cm, and the worst beam has a mean/median distance of around 10 cm. Fig. 10 serves as a visual overview of the point-to-cuboid distances. Each point is colored according to the distance to its cuboid. Violet corresponds to a small, yellow to a large distance. For further examples, we refer to the supplementary material. Fig. 10a offers a view around the inside, with close-up of a cuboid with outliers. The cuboid with the highest mean/median distance is shown in Fig. 10b. Visually, the reconstruction quality of most beams seems to match with Fig. 11.

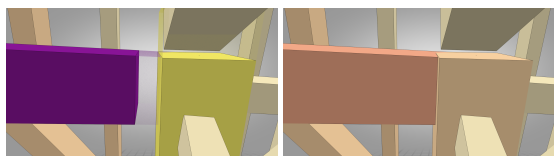
When inspecting the reconstruction result, it becomes apparent that multiple cuboids are cut short despite strong visual evidence that there should be a continuation. This is caused by occlusion and can therefore only be reduced to a certain degree by increasing the effort put into the data acquisition. Thus our method offers a quick user interaction to manually refine the result if desired, as is visualized in Fig. 12. Here, visual inspection revealed a location where two cuboids are not intersecting. Selecting both of them and confirming results in their extrusion, which allows the graph creation to place a vertex there. Additionally, removal of cuboids is necessary if results from inaccurate/wrong planar segmentation *e.g.* in Fig. 8 produces incorrect cuboids. Additionally it is useful to remove cuboids that are valid, however not desirable for a multitude of reasons. *E.g.* in the case of structural analysis, they do not carry any load themselves but introduce load into the system. However, load information can also be added in the simulation software, as demonstrated in Fig. 13. Their removal additionally reduces the computational effort and time necessary for simulating. It is possible to export either the newly created cuboids or the underlying connectivity graph in different formats, *e.g.* for further processing or resuming the task at a later point.

Failure cases for cuboid reconstruction include extraneous or missing cuboids. While we have no remedy for the latter case, the former can be corrected by our interactive editing as explained in Section 3.4.2. Similar to the plane segmentation, a quantitative analysis of the cuboid reconstruction is hard in absence of ground truth data. Thus, we focus on the point-to-respective-cuboid distance in order to analyze the reconstruction quality.





**Figure 11:** Plot of mean/median point-to-cuboid distance. For each cuboid, we compute the mean and median distance of all respective points towards it. The  $x$ -axis denotes the point-to-cuboid distance for the best  $n\%$  of all cuboids. Additionally two red lines, indicate for 50% and 95% of all cuboids their mean/median distance. It is noticeable that only 5% of all cuboids have a larger distance than 3 cm. At the maximum, there are certain beams that have a distance of around 10 cm, which indicates that there exist a handful misaligned beams (cf. Fig. 10b).



**Figure 12:** An example for necessary interactive editing to reduce missing cuboid intersections. A pair of cuboids where visual inspection suggests their intersection is selected manually (left). Refinement confirmation results in the former violet cuboid to be extruded into the yellow one (right).

#### 4.6. Structural Analysis

Changes of use of an existing structure, renovations or even reinforcement often require a recalculation of the structure. This involves capturing the geometric data of the structure in order to generate a FEM-Model (finite element method). Generally, the geometry and cross-sections of all structural components are measured manually for this purpose and then transferred to suitable calculation model. In order to be able to use this function in almost any simulation software, we represent the geometry via an IFC (Industry Foundation Classes) file as an open standard (ISO 16739) in the construction industry, in which each structural component is defined by its axis and cross-section, so that the conversion into a calculation model is possible with the least effort in any software with BIM support.

Our method allows an automated transfer of the obtained scan data into a FEM bar model, which only has to be completed with the parameters required for the calculation (loads, supports and material properties). We achieve this, by first mapping each edge of the connectivity graph to the cuboid it is contained in, this allows us to look up the cross-section of each edge in the graph. Afterwards we can build the IFC file, by creating cross-sections centered at one end of the edge and extrude it along the edge direction until the other end. Each edge represents a beam in the IFC output file.

Fig. 13 (left) shows the calculation model (green) that we generated with our method using the FEM software Infograph. For the simulation, we considered a standard timber with a strength class C24 (according to EN388) applied to the entire model and applied point loads (black arrows) at the ridge points. The structure supports are shown in red. Fig. 13 (right) illustrates the corresponding deformation result. Near the ridge the deformation is 5 mm (red). With increasing distance to the load inputs, more beams are involved in load bearing, resulting in lower deformations until they approach zero near the supports (light blue).

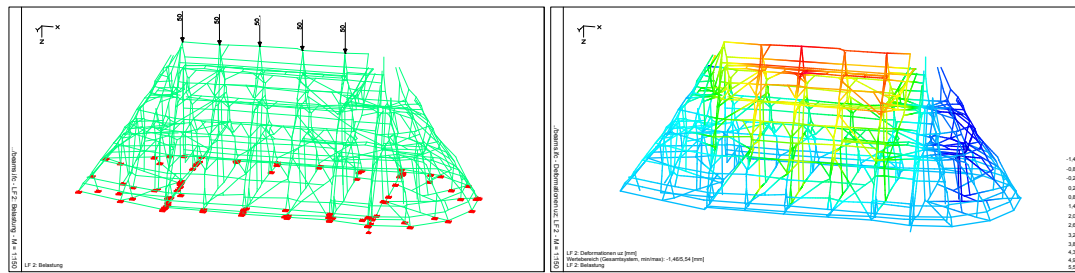
#### 5. Conclusion and Outlook

In this paper we present an automated pipeline that employs a bottom-up feature analysis for obtaining a global structure from wooden beams in scan data. Applications include the digitization of cultural heritage site, *e.g.* the Aachen Cathedral, where information about structural integrity can be inferred directly from scan data. Quality of the reconstruction, as well as viability for structural analysis using commercially available software has been demonstrated. Additionally, we provide improvements over similar algorithms, namely in the planar segmentation and creation of a connectivity graph.

Future work includes improvement on the planar segmentation, as even slight inaccuracies might result in misaligned/incorrect cuboids. It also requires a robust method to distinguish non-dense regions from unions of PRs from each other, as shown in Fig. 8. Additionally, improving the model quality for the structural analysis greatly increases the reliability in the simulation results. A method for reliably establishing cuboid intersections, beyond simple threshold-based intersection tests, would reduce optional interaction for refining the geometric representation.

#### References

[ALOR09] ARMESTO, JULIA, LUBOWIECKA, IZABELA, ORDÓÑEZ, CELESTINO, and RIAL, FERNANDO I. "FEM modeling of structures based



**Figure 13:** Left: FEM overview of the structural analysis setup. All nodes at the bottom marked in red act as support. All black arrows indicate loads added to the system. Here, each arrow contributes 50 kN for a total of 250 kN to the roof ridge points. Right: deformation in vertical direction only for the left image. Red stands for a positive (compression), blue to a negative deformation (decompression). The results should be treated with caution as the model is not precise enough yet to be used for actual decision making.

- on close range digital photogrammetry”. *Automation in Construction* 18.5 (2009), 559–569 2.
- [CDB\*15] CASTELLAZZI, GIOVANNI, D’ALTRI, ANTONIO MARIA, BITELLI, GABRIELE, et al. “From laser scanning to finite element analysis of complex buildings by using a semi-automatic procedure”. *Sensors* 15.8 (2015), 18360–18380 2.
- [DH72] DUDA, RICHARD O and HART, PETER E. “Use of the Hough transformation to detect lines and curves in pictures”. *Communications of the ACM* 15.1 (1972), 11–15 3.
- [EM94] EDELSBRUNNER, HERBERT and MÜCKE, ERNST P. “Three-dimensional alpha shapes”. *ACM Transactions on Graphics (TOG)* 13.1 (1994), 43–72 3.
- [FH75] FUKUNAGA, KEINOSUKE and HOSTETLER, LARRY. “The estimation of the gradient of a density function, with applications in pattern recognition”. *IEEE Transactions on information theory* 21.1 (1975), 32–40 3.
- [Gir16] GIRARDEAU-MONTAUT, DANIEL. “CloudCompare”. *France: EDF R&D Telecom ParisTech* (2016) 1.
- [HZ03] HARTLEY, RICHARD and ZISSERMAN, ANDREW. *Multiple view geometry in computer vision*. Cambridge university press, 2003 2.
- [LPC\*00] LEVOY, MARC, PULLI, KARI, CURLESS, BRIAN, et al. “The digital Michelangelo project: 3D scanning of large statues”. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 2000, 131–144 2.
- [MG20] MURTIYOSO, ARNADI and GRUSSENMEYER, PIERRE. “Virtual disassembling of historical edifices: Experiments and assessments of an automatic approach for classifying multi-scalar point clouds into architectural elements”. *Sensors* 20.8 (2020), 2161 2, 6.
- [MKT21] MOLNÁR, SZILÁRD, KELÉNYI, BENJAMIN, and TAMÁS, LEVENTE. “ToFNest: Efficient normal estimation for time-of-flight depth cameras”. *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 1791–1798 2.
- [MWP18] MURA, CLAUDIO, WYSS, GREGORY, and PAJAROLA, RENATO. “Robust normal estimation in unstructured 3D point clouds by selective normal space exploration”. *The Visual Computer* 34.6 (2018), 961–971 2.
- [ÖPS\*22] ÖZKAN, TAŞKIN, PFEIFER, NORBERT, STYHLER-AYDIN, GUDRUN, et al. “Historic Timber Roof Structure Reconstruction through Automated Analysis of Point Clouds”. *Journal of Imaging* 8.1 (2022), 10 2, 6.
- [PMSK22] POUX, F., MATTES, C., SELMAN, Z., and KOBBELT, L. “Automatic region-growing system for the segmentation of large point clouds”. *Automation in Construction* (2022), 104250. ISSN: 0926-5805. DOI: <https://doi.org/10.1016/j.autcon.2022.104250> 3.
- [PSDP17] PÖCHTRAGER, MARKUS, STYHLER-AYDIN, G, DÖRING-WILLIAMS, MARINA, and PFEIFER, NORBERT. “Automated Reconstruction of historic roof structures from point clouds—development and examples.” *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences* 4 (2017) 6.
- [PSDP18] PÖCHTRAGER, MARKUS, STYHLER-AYDIN, GUDRUN, DÖRING-WILLIAMS, MARINA, and PFEIFER, NORBERT. “Digital reconstruction of historic roof structures: Developing a workflow for a highly automated analysis”. *Virtual Archaeology Review* 9.19 (2018), 21–33 6.
- [RCM\*01] ROCCHINI, CMPPC, CIGNONI, PAULO, MONTANI, CLAUDIO, et al. “A low cost 3D scanner based on structured light”. *computer graphics forum*. Vol. 20. 3. Wiley Online Library. 2001, 299–308 2.
- [Rem03] REMONDINO, FABIO. “From point cloud to surface: the modeling and visualization problem”. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 34 (2003) 1.
- [THA\*10] TANG, PINGBO, HUBER, DANIEL, AKINCI, BURCU, et al. “Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques”. *Automation in construction* 19.7 (2010), 829–843 2.
- [The22] THE CGAL PROJECT. *CGAL User and Reference Manual*. 5.4. CGAL Editorial Board, 2022. URL: <https://doc.cgal.org/5.4/Manual/packages.html> 3.
- [VAC12] VALERO, ENRIQUE, ADAN, ANTONIO, and CERRADA, CARLOS. “Automatic construction of 3D basic-semantic models of inhabited interiors using laser scanners and RFID sensors”. *Sensors* 12.5 (2012), 5705–5724 2.
- [VSS14] VOLK, REBEKKA, STENGEL, JULIAN, and SCHULTMANN, FRANK. “Building Information Modeling (BIM) for existing buildings—Literature review and future needs”. *Automation in construction* 38 (2014), 109–127 2.
- [WXW18] WU, QIAOYUN, XU, KAI, and WANG, JUN. “Constructing 3D CSG models from 3D raw point clouds”. *Computer Graphics Forum*. Vol. 37. 5. Wiley Online Library. 2018, 221–232 2.
- [XAAH13] XIONG, XUEHAN, ADAN, ANTONIO, AKINCI, BURCU, and HUBER, DANIEL. “Automatic creation of semantically rich 3D building models from laser scanner data”. *Automation in construction* 31 (2013), 325–337 2.
- [XTHS18] XU, YUSHENG, TUTTAS, SEBASTIAN, HOEGNER, LUDWIG, and STILLA, UWE. “Reconstruction of scaffolds from a photogrammetric point cloud of construction sites using a novel 3D local feature descriptor”. *Automation in Construction* 85 (2018), 76–95 2.
- [YKG17] YANG, X, KOEHL, M, and GRUSSENMEYER, P. “Parametric modelling of as-built beam framed structure in BIM environment”. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 42 (2017), 651 2.