

**humantech**

## **D3.6 – Complete Scan-to-BIMxD Digital Twin pipeline**



This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement n° 101058236. This document reflects only the author's view, and the EU Commission is not responsible for any use that may be made of the information it contains.



### D3.6 – Complete Scan-to-BIMxD Digital Twin pipeline

<b>Project Title</b>	Human-Centred Technologies for a Safer and Greener European Construction Industry.
<b>Project Acronym</b>	HumanTech
<b>Grant Agreement No</b>	101058236
<b>Instrument</b>	Research & Innovation Action
<b>Topic</b>	HORIZON-CL4-2021-TWIN-TRANSITION-01-12
<b>Start Date of Project</b>	June 1, 2022
<b>Duration of Project</b>	36 months

<b>Name of the Deliverable</b>	Complete Scan-to-BIMxD Digital Twin pipeline
<b>Number of the Deliverable</b>	D3.6 (D15)
<b>Related WP Number and Name</b>	<b>WP3 - Dynamic Semantic Digital Twin Generation</b>
<b>Related Task Number and Name</b>	T3.6 From 3D scans to BIMxD representations
<b>Deliverable Dissemination Level</b>	Public
<b>Deliverable Due Date</b>	M29
<b>Deliverable Submission Date</b>	31.10.2024
<b>Task Leader/Main Author</b>	Fabian Kaufmann, RPTU
<b>Contributing Partners</b>	DFKI
<b>Reviewer(s)</b>	Jason Rambach, Marius Schellen



### Keywords

Scan-to-BIM

## Revisions

Version	Submission date	Comments	Author
v0.1			
v0.2			
...			
...			
...			

## Disclaimer

This document is provided with no warranties whatsoever, including any warranty of merchantability, non-infringement, fitness for any particular purpose, or any other warranty with respect to any information, result, proposal, specification, or sample contained or referred to herein. Any liability, including liability for infringement of any proprietary rights, regarding the use of this document or any information contained herein is disclaimed. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by or in connection with this document. This document is subject to change without notice. HumanTech has been financed with support from the European Commission. This document reflects only the view of the author(s) and the European Commission cannot be held responsible for any use which may be made of the information contained.



## Acronyms and definitions

Acronym	Meaning
14NH	14 Nothelfer hospital Weingarten
BIM	Building Information Modeling
CV4AEC	Computer vision in the built environment
IoU	Intersection over Union

### Abstract

This deliverable reports the development of a complete Scan-to-BIMxD pipeline delivered by the HumanTech project. Initially, a semantic segmentation of the point cloud is performed followed by geometric reconstruction procedures. Open BIM authoring tools are then used to deliver BIM objects of wall, door and columns. Finally, results of our experiments will be reported.



## The HumanTech project

The European construction industry faces three major challenges: increase the safety and wellbeing of its workforce, improve its productivity, and become greener, making efficient use of resources.

To address these challenges, HumanTech proposes to develop **human-centred cutting-edge technologies** such as wearables for workers' safety and support and robots that can harmoniously coexist with human workers while contributing to the ecological transition of the sector.

**HumanTech aims to achieve major advances in cutting-edge technologies that will enable a safe, rewarding, and digital work environment for a new generation of highly skilled construction workers and engineers.**

These advances will include:

- **Robotic devices equipped with vision and intelligence** that allow them to navigate autonomously and safely in highly unstructured environments, collaborate with humans and dynamically update a semantic digital twin of the construction site in which they are.
- **Smart, unobtrusive workers protection and support equipment.** From exoskeletons activated by body sensors for posture and strain to wearable cameras and XR glasses that provide real-time workers' location and guidance for them to perform their tasks efficiently and accurately.
- An entirely new breed of **Dynamic Semantic Digital Twins (DSDTs) of construction sites** that simulate in detail the current state of a construction site at the geometric and semantic level, based on an extended Building Information Modelling (BIM) formulation that contains all relevant structural and semantic dimensions (BIMxD). BIMxDs will act as a common reference for all human workers, engineers, and autonomous machines.

The **HumanTech consortium** is formed by 22 organisations — leading research institutes and universities, innovative hi-tech SMEs, and large enterprises, construction groups and a construction SME representative — from 10 countries, bringing expertise in 11 different disciplines. The consortium is led by the German Research Center for Artificial Intelligence's Augmented Vision department.



# Contents

1	Introduction.....	8
1.1	Motivation and Scope.....	8
1.2	Methodology .....	9
1.3	Scan to BIMxD in the HumanTech framework.....	12
2	Semantic segmentation of point clouds .....	13
3	Geometric feature extraction.....	13
3.1	Introduction .....	13
3.2	Density based clutter removal and voxel downsampling .....	14
3.3	Levels and storeys .....	16
3.4	Walls reconstruction .....	18
3.4.1	Axis sorting of walls .....	18
3.4.2	Instance segmentation .....	20
3.4.3	Plane fitting.....	21
3.4.4	Bounding box fitting.....	22
3.5	Doors reconstruction .....	23
3.6	Columns reconstruction.....	26
4	<i>Pystruct3d</i> : efficient geometry reconstruction algorithms .....	27
4.1	<i>bbox</i> : Fitting and manipulating bounding boxes.....	28
4.2	<i>visualization</i> : visualize bounding boxes and input data .....	29
4.3	<i>metrics</i> : evaluating the reconstruction accuracy .....	30
5	Open source BIM authoring .....	30
5.1	From bounding box to IFC geometry.....	30
5.2	Semantic IFC object creation.....	33
5.2.1	Material information.....	33
5.2.2	Property sets .....	33
5.2.3	Spatial relationships.....	33
6	Experiments and results.....	34



## D3.6 – Complete Scan-to-BIMxD Digital Twin pipeline

---

6.1	CV4AEC challenge.....	34
6.2	14 Nothelfer hospital, Weingarten, Germany.....	35
6.3	KPIs.....	37
7	Discussion and conclusions.....	37

# 1 Introduction

## 1.1 Motivation and Scope

Naturally, BIMs are typically not available for existing structures. Thus, it seems reasonable to create BIMs out of existing structures. This procedure is referred to as *scan-to-BIM*. Within this process, 3D data to create the semantic models is 'scanned', i.e. acquired using reality capturing devices. However, scan-to-BIM procedures were and still are far from being automated, and a manual process partially supported by software workflows.

Regarding the technology level, significant advances have been witnessed in recent years, namely in the fields of semantic segmentation, geometry processing algorithms and automating scan-to-BIM itself. Fusing the most significant advances in all three fields can leverage the automation of scan-to-BIM procedures. Thus, the main contribution of this work will be to provide a comprehensive pipeline for scan-to-BIMxD automation to support the HumanTech technology framework. In this context, BIMxD refers to an extended BIM formulation based on IFC. The specific contributions can be outlined as follows:

- To train semantic segmentation models, manually annotated training data is required. Therefore, this work will contribute to this field with comprehensive data annotation guidelines, a workflow to support this manual process and a dataset to leverage semantic segmentation.
- In a joint effort of RPTU and the Department of Augmented Vision at the German Research Centre for Artificial Intelligence (DFKI), a semantic segmentation model was trained using the previously annotated dataset.
- Several algorithms useful for geometric reconstruction have been developed and implemented to offer the foundation for comprehensive reconstruction pipelines.
- Likewise, a module was developed to combine the previously reconstructed geometry with semantic information and to create BIM objects using an open data standard.
- The procedures are combined to an end-to-end pipeline, which is extensively tested.
- Beyond the well-known evaluation metrics, enhanced metrics have been developed and implemented. Along with ground truth BIMs and aligned point clouds, these metrics serve to assess the quality of the reconstruction.

## 1.2 Methodology

The basis for this work is an aligned and complete 3D point cloud of the object with a sufficient density. In this context, a spatial resolution of 5mm or higher can be considered sufficient i.e., any physical object should be covered by scan points with a maximum distance of 5mm. For parts of the framework such as semantic segmentation, colour values are mandatory to distinguish objects reliably. Scan-to-BIMxD pipeline is developed sensor-agnostic, i.e. point clouds captured with various sensors can be processed. However, the raw scans need to be aligned so that all scans share a common coordinate frame. Depending on the purpose of the BIM model created, a translation into a global coordinate frame may be required, e.g. to compare the acquired model with an existing model or to generally use it in conjunction with other data or BIMs. Typically, the referencing is achieved using conventional surveying techniques providing global coordinates of markers placed before the scanning. The data acquisition and alignment itself is not part of this work. Further information regarding scanning technology can be found in D 3.1.

Considering the accuracy of the input data, we assume that the acquired point cloud accurately represents the physical objects. Clearly, this might not apply as absolute measurement errors and inaccuracies during the alignment can influence the overall accuracy. Nevertheless, the input point clouds are even used to evaluate the resulting BIMs. Therefore, errors resulting from the data acquisition are excluded from the evaluation, thus presenting unambiguous evaluations of the results.

It is commonly observed that acquired point clouds of buildings are incomplete to a certain extent. The incompleteness depends on the scanning technology applied and the effort spent to acquire a point cloud as complete as possible. However, incompleteness occurs and results in:

- Partially scanned objects, i.e. only the surfaces visible to the scanner have been captured.
- Limitations in range of the acquisition system.
- Limited accessibility, e.g. of outside surfaces at high elevation for terrestrial scanner.
- Objects occluding others, resulting in missing parts of objects' surfaces.
- Partially missing data due to over exposure to natural light, shadows or insufficient lighting conditions.
- Translucent objects, e.g. glass panes not reflecting light waves.
- Highly reflective or mirror surfaces resulting in scanning artefacts.
- Light absorbing objects

In the Scan-to-BIMxD pipeline, methods developed are tailored to deal with incomplete data, thus achieving the most accurate reconstruction possible. If these methods still fail, predefined parameters



are applied during the reconstruction, e.g. a predefined wall thickness in case only one vertical surface of a wall is present in the data. Likewise, it must be expected, that during the processing, data points can get lost, e.g. due to false classification in the semantic or geometric segmentation procedures applied. Thus, strategies will be implemented and evaluated to handle incomplete segmentation, still providing an accurate and complete BIM reconstruction.

During the reconstruction, the Manhattan world assumption is applied<sup>1</sup>. In a Manhattan world scenario, any object is aligned with a Cartesian coordinate system. Considering typical buildings, this accounts for a vast majority of the objects observed. To apply the Manhattan assumption, the data must be axis-aligned, which can be achieved manually with little effort. The transformation can be applied inversely to transform the reconstructed model back to a global coordinate frame.

---

<sup>1</sup> James M. Coughlan and Alan L. Yuille. "Manhattan world: Compass direction from a single image by bayesian inference". In: Proceedings of the seventh IEEE international conference on computer vision. Vol. 2. 1999, pp. 941–947.

Based on the constraints and assumptions that have been described, a framework for automated scan-to-BIM pipelines is developed. This framework includes semantic and instance segmentation, geometry reconstruction and BIM model creation (refer to (I-V) in Figure 1).

Semantic segmentation is referred to as the method to assign a class label to every point in the point cloud (refer to (II) in Figure 1). In the context of this work, a building component label is assigned. The semantic segmentation model was developed jointly in the HumanTech project by Rambach, Chamseddine and Kaufmann<sup>2</sup>. This work contributes to the semantic segmentation by providing a comprehensive annotation guideline that introduces civil engineering expertise into data annotation. This leads to technically sound annotated data compliant with the structure of BIMs along with data annotated according to these guidelines.

The next step is instance segmentation, i.e. segmenting the set of points of one class into individual objects (see Figure 1 (III)). Hereby, point axis sorting, clustering and primitive fitting algorithms are applied to cluster, segment and group per-object points. As a side effect, the methods apply a filtering to the instances, thus improving the accuracy of the reconstruction.

The backbone of geometric reconstruction is a tailored bounding box fitting algorithm to retrieve the transformation, translation and rotation based on the previously segmented instances (Figure 1 (IV)). By applying certain assumptions the algorithm can be reduced to a 2.5D problem, i.e. the boundaries can be reconstructed from a 2D projection of the points. Before applying the bounding box reconstruction, the shape of the objects is determined. For occurrences of round shaped objects, a robust cylinder primitive fitting algorithm is applied. Some refinement procedures may be applied to increase the quality of the reconstruction, including modifying objects when exceeding dimension limits and applying topology correction to close gaps between adjacent perpendicular objects or to extend / clip objects that share the same centre line.

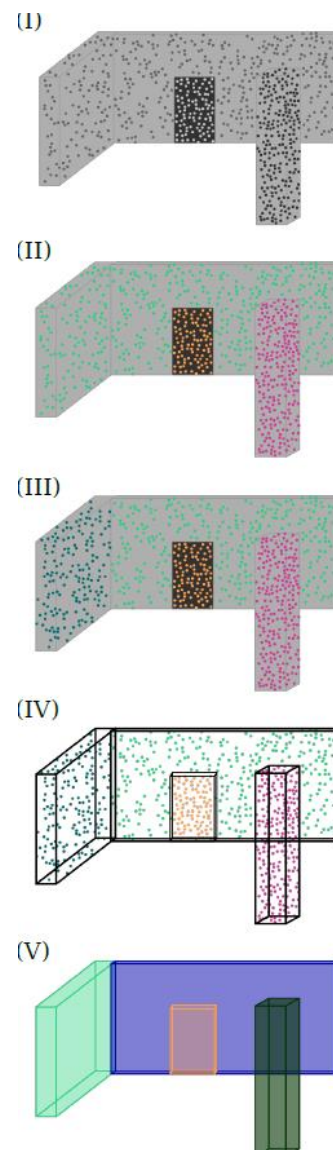


Figure 1 Pipeline overview

<sup>2</sup> Mahdi Chamseddine, Jason Rambach, and Fabian Kaufmann. “BIMStruct3D: A fully automated hybrid learning Scan-to-BIM pipeline with watertight topology reconstruction: [Manuscript in preparation]”.

From the refined reconstruction, all parameters to create BIM objects can be retrieved. The objects are created in the open Industry Foundation Classes (IFC) format<sup>3</sup>, thus maximising interoperability of the reconstructed models. To create IFC objects, BonsaiBIM (BlenderBIM) and IfcOpenShell<sup>4</sup> in combination with the HumanTech open-source BIM authoring tools (refer to Deliverable 2.2) were used, which offer comprehensive libraries for manipulating IFC data. Being open source and an open ecosystem, these libraries allow for future scaling of the method proposed here.

### 1.3 Scan to BIMxD in the HumanTech framework

The Scan to BIMxD pipeline is deeply integrated into the HumanTech architecture and framework (refer to Deliverables 1.2 and 1.4 for details). Thus, it is linked to all pilots. The scan to BIMxD pipeline is or will be deeply integrated into the DSDT pilots 1 and 4, the former related to building and construction processes, the latter related to bridge monitoring and capturing.

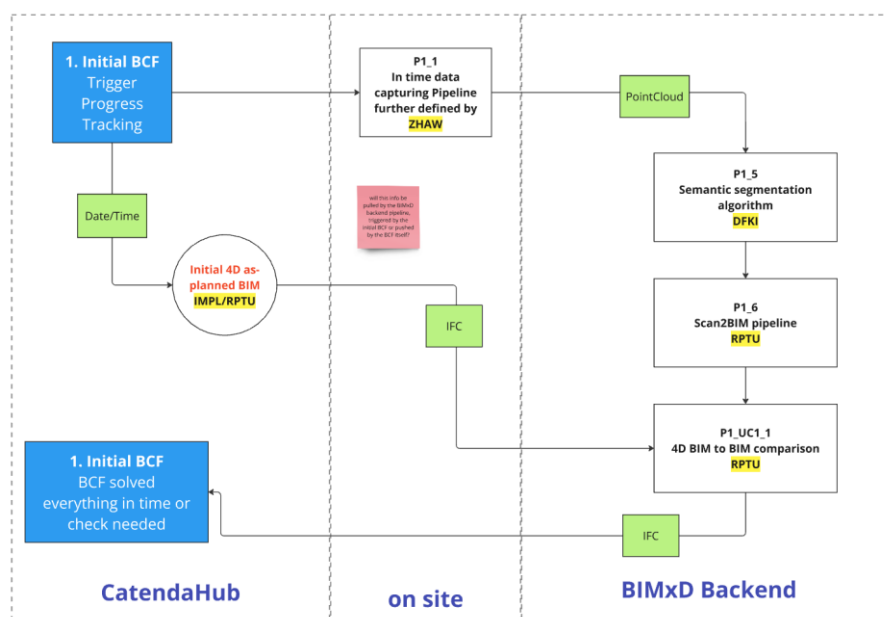


Figure 2 Pilot 1 use case 1 construction progress monitoring diagram. P1\_6 is based on the Scan to BIMxD digital twin pipeline and fully integrated to deliver the pilot scenario.

Figure 2 shows the flow diagram of Pilot 1 Use case 2 “Construction progress monitoring”. Beyond the data acquisition, a full BIM reconstruction using the scan to BIMxD pipeline (P1\_6 in Figure 2) will serve as the intermediate representation to track the progress. The progress itself will be monitored by comparing the currently reconstructed BIM with the former one. Likewise, the other pilot use cases will employ the scan to BIMxD pipeline or at least parts of it. Beyond that, the scan-to-BIMxD pipeline is well integrated with the developed scanning technology and BIMxD backend.

Implicitly, Scan to BIMxD is a core part of even the robotic applications in HumanTech. As most of these applications rely on a BIM for task planning, it needs to be reconstructed if not yet available.

<sup>3</sup> Buildingsmart. Industry Foundation Classes: Version 4.2 bSI Draft Standard IFC Bridge proposed extension. Ed. by Buildingsmart. 2019. URL: [https://standards.buildingsmart.org/IFC/DEV/IFC4\\_2/FINAL/HTML/](https://standards.buildingsmart.org/IFC/DEV/IFC4_2/FINAL/HTML/) (visited on 10/01/2022).

<sup>4</sup> Thomas Krijnen. IfcOpenShell. 2021. URL: <http://ifcopenshell.org/>.

## 2 Semantic segmentation of point clouds

The scan-to-BIM pipeline uses the Semantic Segmentation methods as described in D3.5. Details can be found in this Deliverable. For the scan-to-BIM semantic segmentation is the first step, followed by geometric reconstruction and BIM delivery.

## 3 Geometric feature extraction

### 3.1 Introduction

The input for geometric feature extraction is a segmented point cloud. With the semantic labels, all points of a specific component class can be extracted from the point cloud and processed individually. Thus, tailored feature extraction strategies can be developed and applied. Yet, these strategies share the following common sequence of steps:

- **Clutter filtering:** Noise and clutter outside the building perimeter are the result of objects scanned through openings of the building, reflections from translucent objects, vegetation outside the building, objects in the field of view of the scanner and other artefacts. In actual point clouds of buildings such clutter is typically present and should be removed as one of the first steps to avoid reconstruction artefacts. Automated clutter filtering can be applied to the entire set of input points or to a subset of points of a particular class.
- **Instance segmentation:** During semantic segmentation, class labels are assigned to all points. Through filtering, all points of a component class can be extracted from the point cloud. However, individual objects cannot be distinguished. Thus, a method for instance segmentation needs to be applied. To achieve this, clustering strategies exploiting the object orientation, point density and relation to parent objects are developed and used. From this step, points of one object instance are retrieved.
- **Parameter extraction:** Per object instance, the geometric parameters are extracted including shape, dimension, translation and rotation. This can involve one or multiple steps of primitive fitting, filtering and bounding box fitting. For cuboid shaped objects, a bounding box defined by eight corner points is used to represent the result. For other shapes, similar representations can be defined.

Considering that any of the reconstruction tasks is a 3D problem, at least 9 Degrees of Freedom (DoF) (3D translation, 3D rotation, 3D dimension) need to be solved. If the type of primitive applying to an object is unknown, this adds one more degree of freedom to the problem. Beyond the mere dimensions, the general shape of the object first needs to be defined. However, objects might not only consist of one primitive shape but might be a combination of primitive shapes or topologically even more complex.

Developing geometric feature extraction algorithms that can be applied in a very general manner, i.e. on arbitrary shaped objects without any constraints of orientation, is a complex task. Such algorithms are likely to be computationally expensive, too. From a practical point of view, it can even be argued that an efficient reconstruction algorithm is more useful than a general yet inefficient one, provided that uncertain reconstructions raise an error that can be handled in a subsequent processing step or even manually. Thus, it seems desirable to simplify the reconstruction problem whenever this is sensible from a practical point of view.

Being formulated by Coughlan et al.<sup>5</sup>, the Manhattan-world-assumption states that in an urban context, most objects are aligned along a two-dimensional *Manhattan* grid, i.e. along axes perpendicular to each other. Being a reminiscence to Coughlan's work on determining the user orientation with a Computer Vision algorithm, the term became a synonym for any objects aligned with a 2D perpendicular grid. This notion also applies to most buildings themselves, where most objects are aligned along a set of perpendicular axes. Whenever the axes are parallel to the axes of a Cartesian coordinate system, the data can be considered axis-aligned. In this work, not only the Manhattan-world-assumption is used to simplify the reconstruction procedures, as even the input data needs to be axis-aligned. Although this is not the case in real-world scenes, the input point cloud can either be axis-aligned programmatically using Principal Component Analysis (PCA) or manually with little effort.

Beyond that, the reconstruction problem can be simplified by defining constraints for object classes, which apply for the majority of the objects observed. The constraints can include shape or primitive, planarity, horizontal or vertical alignment, orientation of bounding surfaces, etc.

In the following sections, the methods will be described along the reconstruction sequence of components, starting with walls, continuing with doors being child objects of walls and ending with columns representing typical members of structure skeletons. Although some concepts apply to several or all components to be reconstructed, it seems more natural to explain the process sequentially rather than elucidating on a theoretical level with only a minor relation to the actual reconstruction stack. Some more sections will be dedicated to storey reconstruction and clutter filtering.

## 3.2 Density based clutter removal and voxel downsampling

In a first attempt to reduce the number of points in the point cloud, a density-based clutter removal algorithm is developed and applied to clean the set of points from those that are not of interest to the reconstruction. Clutter points in this case include scanning artefacts outside the building caused by

---

<sup>5</sup> James M. Coughlan and Alan L. Yuille. "Manhattan world: Compass direction from a single image by bayesian inference". In: Proceedings of the seventh IEEE international conference on computer vision. Vol. 2. 1999, pp. 941–947.

reflecting or translucent surface deflections, vegetation, moving vehicles and pedestrians, neighbouring buildings, etc.

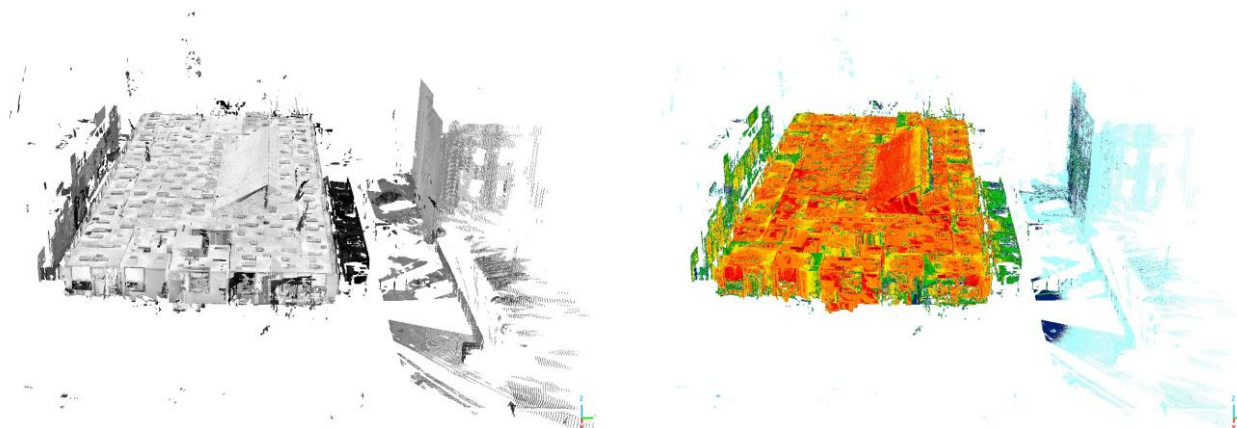


Figure 3 Lower density of clutter points. Left: Input point cloud. One storey of an office building in the centre surrounded by artefacts and clutter. Right: Point density calculated by determining the number of neighbours in a given radius of 2.5 cm. The density is visualised using a colour scale spanning from blue (low density) to red (maximum density).

The core observation that motivated the development of this density-based clutter removal algorithm is that clutter points, especially those outside of the building, have a lower point density than the object of interest itself (see Figure 3). As the point density relates to the distance of the object captured, objects of interest related to the building at a closer distance are retrieved with a higher point density and clutter points scanned from a farther distance with a lower point density. This fact can be used to remove clutter based on its lower density compared to building elements of interest.

The density-based clutter removal algorithm used in this work includes the following procedure. First, a voxel grid is built on the point cloud. To identify sparsely or not populated voxels, the number of points inside a particular voxel is calculated. All voxels populated with a number of points below the predefined limit will be considered as clutter. The inliers, i.e. points of interest, are then identified by constructing a mask with indices of inlying points that is applied on the input wall point cloud. The result of this filtering step is presented in Figure 4 left. The orange bounding box encloses the points of the building to be reconstructed. Note that inside the bounding box points were considered as clutter (grey colour) by the algorithm.

To avoid discarding points with a lower density from within the building itself, the minimum oriented bounding box of all inlier points is calculated using A Modern Library for 3D Data Processing (Open3D)'s *get minimal oriented bounding box* method, which provides the bounding box with the smallest volume<sup>6</sup>. All points inside this bounding box are considered for further processing thus removing low density clutter outside the perimeter of the building but preserving low dense details.

<sup>6</sup> Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. "Open3D: A Modern Library for 3D Data Processing". In: arXiv:1801.09847 (2018).

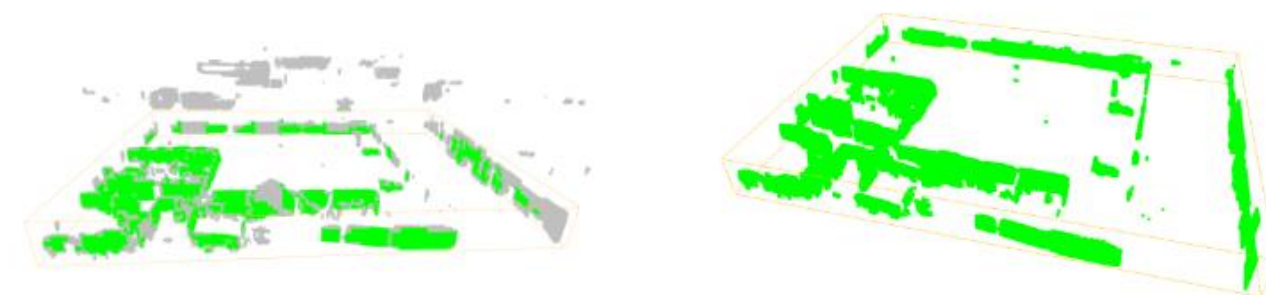


Figure 4 Result of the density based clutter removal algorithm.

In the example given in Figure 4 Result of the density based clutter removal algorithm. the result of the density based clutter removal algorithm as a set of wall points was processed. A voxel size 0.65 m was chosen, and the density limit, i.e. minimum number of points per voxel, was set to 2500. Depending on the density of the input point cloud, and downsampling applied before, these parameters need to be adjusted. A disadvantage of the proposed method is that it is computationally expensive and requires huge memory if a small voxel size is chosen. For point clouds of buildings, a large voxel size of more than 0.5 m is acceptable and provides good filtering results at a reasonable computation time.

As this algorithm exploits the point density, it is crucial to apply it to the input point cloud before any other downsampling procedures, which could potentially harmonise the point density. Thus, voxel downsampling procedures are then applied depending on the point density of the input data to harmonise the density and reduce the number of points ensuring efficient processing.

### 3.3 Levels and storeys

Many elements are bounded by horizontal structural components, such as slabs, or by floor and ceiling when the structural element is concealed. In BIM models, objects are typically associated with levels to organise the model. Identifying these levels is thus a fundamental initial step in facilitating the reconstruction of other elements.

Level reconstruction algorithms can be designed and applied either to the entire point set without prior semantic segmentation, or specifically to points identified as slab, floor, and ceiling. When semantic segmentation is applied beforehand, all points categorised as floor, ceiling, and slab are grouped horizontally. The average of all z-coordinate values within each horizontal group is then calculated to determine the heights of the floor, ceiling, or slab. The vertical distances allow for the identification of the floor or the top surface of the slab.

In case the complete input point cloud is used, peaks in the z-coordinate value distribution are analysed. Although being computationally more expensive, this method offers the benefit of not relying on the accuracy of semantic segmentation that may be incomplete resulting in errors in the level heights computed. Additionally, inaccuracies in determining level heights can significantly impact every object

reliant on the horizontal constraint hypothesis. Thus, it is recommended to use the entire input point cloud. In previous research, similar approaches to horizontal segmentation based on a point density histogram have been introduced in<sup>7,8</sup> and. The algorithm proposed here is based on these approaches. However,<sup>9</sup> use a point density histogram to split the point cloud in between floor and ceiling. Likewise,<sup>10</sup> use the point density along the z-axis to split the point cloud into storeys to then further divide it into subspaces such as rooms. Although the algorithm proposed here uses a similar approach, it is designed to identify the level heights as an input for further object reconstruction.

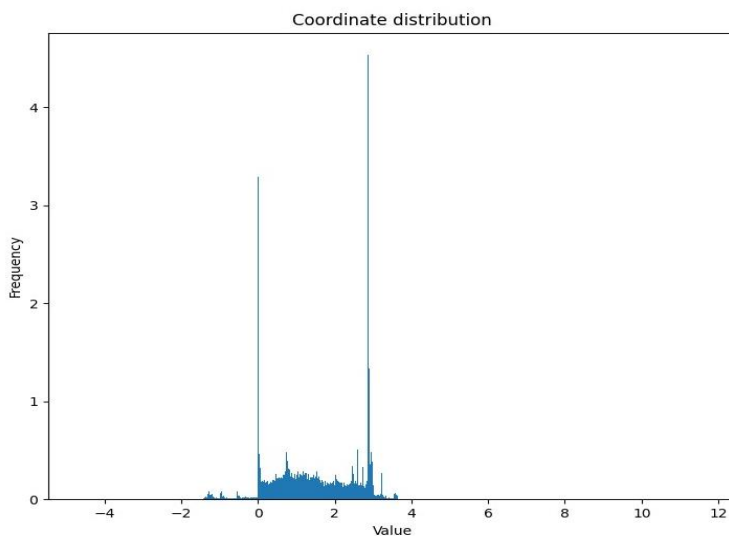


Figure 5 Distribution of z coordinates of a one-storey building. As expected, two peaks appear at the level heights.

The basic principle of finding levels is to exploit the point distribution of points in vertical direction (Z-axis). Intuitively, horizontal floor, ceiling or slab surfaces should cause a remarkable concentration of points in vertical direction. This phenomenon can be observed in Figure 5 which shows the point distribution along the Z-axis of a one-storey office building. The centre bins of the histogram (from 0 to 3) represent most of the objects (structural elements, furniture, other

objects, clutter). The two major peaks represent the floor and ceiling, as for horizontal surfaces the number of points with equal z coordinate value will be significantly higher than the relatively homogeneous z distribution in between.

The procedure chosen can be described as follows. Given a point cloud, a histogram is defined by partitioning the data set into k bins. In this case, only the Z component of the point coordinates is

<sup>7</sup> Maarten Bassier, Meisam Yousefzadeh, and Maarten Vergauwen. “Comparison of 2D and 3D wall reconstruction algorithms from point cloud data for as-built BIM”. In: Journal of Information Technology in Construction 25 (2020), pp. 173–192. DOI: 10.36680/j.itcon.2020.011.

<sup>8</sup> Hélène Macher, Tania Landes, and Pierre Grussenmeyer. “From Point Clouds to Building Information Models: 3D Semi-Automatic Reconstruction of Indoors of Existing Buildings”. In: Applied Sciences 7.10 (2017), p. 1030. DOI: 10.3390/app7101030.

<sup>9</sup> Maarten Bassier, Meisam Yousefzadeh, and Maarten Vergauwen. “Comparison of 2D and 3D wall reconstruction algorithms from point cloud data for as-built BIM”. In: Journal of Information Technology in Construction 25 (2020), pp. 173–192. DOI: 10.36680/j.itcon.2020.011.

<sup>10</sup> Hélène Macher, Tania Landes, and Pierre Grussenmeyer. “From Point Clouds to Building Information Models: 3D Semi-Automatic Reconstruction of Indoors of Existing Buildings”. In: Applied Sciences 7.10 (2017), p. 1030. DOI: 10.3390/app7101030.

considered. Identifying peaks can be formulated as finding local maxima in the histogram. A bin is considered a peak if its count is greater than the counts of its neighbours.

Floor and ceiling or upper and lower surface of slabs can hereby be distinguished by the distance of the respective levels and prior defined thresholds such as the minimum room height or the maximum vertical distance of floor and ceiling. By applying these constraints, false positive level detection can be avoided.

Having reconstructed the levels, the object reconstruction can be initiated starting with the walls.

### 3.4 **Walls reconstruction**

Beyond the vertical organization of a building by its levels, walls are the main objects that enclose and separate the spaces thus forming the topology of the building. Hence, it is sensible to reconstruct the walls after the levels and storeys to provide these elements for the subsequent reconstruction of child objects of walls, e.g. doors, windows, openings, etc.

The core principle of wall reconstruction is to perform a per-direction reconstruction of the walls, followed by instance segmentation including clustering, plane fitting and parallel plane grouping. A minimum bounding box is then fitted to the wall instances.

These operations are performed on the subset of points assigned with the label wall. Hence, the first procedure is to filter the subset of a given label from the input points. Instead of iterative checking each point's label, a mask of boolean values is created from the labels array, where True indicates all positions with a given label. Applying this mask to the points array is an efficient way to retrieve all points of the given label.

#### 3.4.1 **Axis sorting of walls**

Reconstructing walls direction-wise yields the main benefit, that intersections at wall corners are eliminated from the per-axis data, thus resulting in a more reliable instance segmentation. Parallel walls are typically separated by at least the distance accessible for persons. This fact can be used to apply density-based algorithms more efficiently and reliably.

Based on the Manhattan-world assumption<sup>11</sup> and the initial axis-alignment of the data, the point normals can be used to sort the walls according to their respective direction, namely walls oriented parallel to the X or Y-axis.

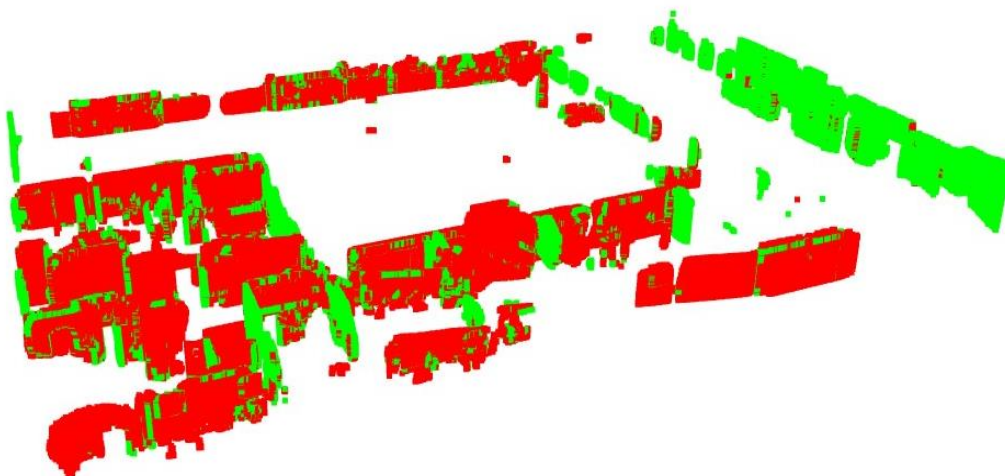
---

<sup>11</sup> James M. Coughlan and Alan L. Yuille. "Manhattan world: Compass direction from a single image by bayesian inference". In: Proceedings of the seventh IEEE international conference on computer vision. Vol. 2. 1999, pp. 941–947.

The axis sorting algorithm includes two steps: (i) calculating the point normals, i.e. the vector normal to a point considering several adjacent points and (ii) splitting the set of points into two subsets of such points normal to the X or Y direction, respectively.

First, the normals are calculated within a local neighbourhood of points. The points within the local neighbourhood are identified by a k-nearest-neighbour search using the Open3D implementation<sup>12</sup>. The maximum number of points is limited to facilitate fast normal computation. The normal itself is computed using the Eigenvalues of the covariance matrix, with the third Eigenvalue equivalent to the point normal. This normal value is then assigned to the respective point. The algorithm outputs two normal vectors of opposite direction, which both can be correct. Without any context, the direction is randomly chosen. This behaviour is acceptable, as only the orientation is needed for axis sorting, not the actual direction of the point normal.

To sort all walls that are close or in x-axis direction, all points with normals' Y component equal to one within a predefined limit of  $\pm 0.1$  are selected. The procedure is applied for Y direction, respectively. The normal vectors should be normalised before this operation, i.e. the length of the normal vector should equal 1. There is no strict verticality constraint applied but implicitly, points of non-vertical direction are not considered as such arbitrary orientation of the point normal causes the X or Y component of the normal vector to drop below the limit. An example of wall points sorted in axis direction is presented in Figure 6.



*Figure 6 Axis sorted walls.*

---

<sup>12</sup> Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. "Open3D: A Modern Library for 3D Data Processing". In: arXiv:1801.09847 (2018).

### 3.4.2 Instance segmentation

From the axis sorted points, wall instances need to be segmented. Based on the analysis of basic geometry processing algorithms, plane fitting and density-based clustering are the alternatives to be considered. Although plane fitting algorithms robust to noise exist, we experienced limitations of applying algorithms such as RANSAC<sup>13</sup> directly to a larger set of points. One of the main limitations is that the input parameters need to be selected carefully, which is difficult on an unseen dataset in an automated scenario. Some insights into the limitations of RANSAC will be given in the next section, revealing that such procedures applied to a larger set of points do not yield robust and accurate results. This leads to the preference of density-based clustering algorithms for three reasons. (i) In a scenario with spatially separated clusters as is the case with axis sorted wall points, density-based clustering algorithms such as DBSCAN introduced by Ester et al.<sup>14</sup> yield robust and accurate results with only a few, easy-to-set input parameters. Indeed, the parameters could be almost set globally, with tiny tweaks to reduce the processing time. (ii) The algorithm only returns an instance label per point, no further information such as primitive parameters. Although this might sound like a disadvantage, it is in fact irrelevant as such parameters would not be used at this stage. (iii) DBSCAN does not require the number of clusters as an input, which would be impractical to select on unseen data.

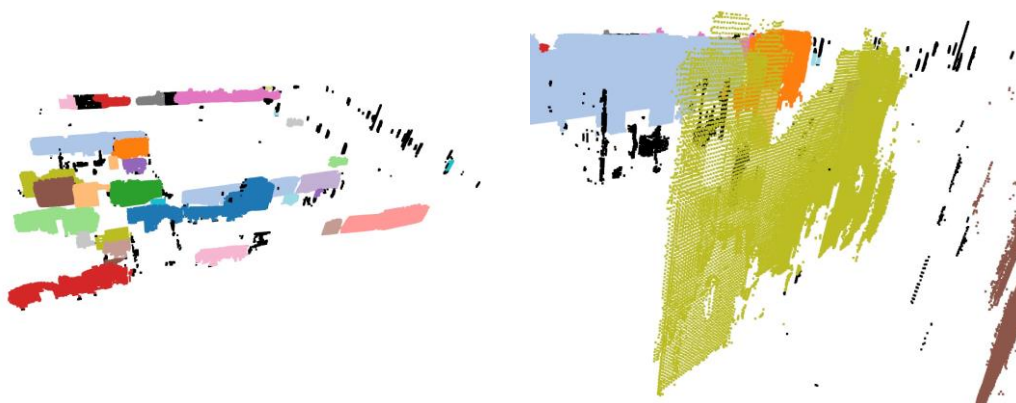


Figure 7 DBSCAN clustering. Left: set of wall points with clusters visualised. Right: Detail of clustered walls with noise and clutter. Coloured points are assigned to a cluster, black points are considered as clutter.

An example of a batch of axis-sorted wall points that the DBSCAN algorithm was applied to is shown in Figure 7. All clusters are coloured randomly. Note that colours might repeat as the number of colours in the colour map is limited. Points that were not assigned to a cluster are black.

<sup>13</sup> Ruwen Schnabel, Roland Wahl, and Reinhard Klein. “Efficient RANSAC for Point-Cloud Shape Detection”. In: Computer Graphics Forum 26.2 (2007), pp. 214–226.

<sup>14</sup> Martin Ester et al. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. KDD’96. AAAI Press, 1996, pp. 226–231.

### 3.4.3 Plane fitting

The scope of wall reconstruction has been contained to walls with planar surfaces. As this limitation applies to the vast majority of wall objects observed in building point cloud data, it is sensible to try to reconstruct the planar surfaces of the wall instances observed. To this end, plane fitting algorithms can be utilized. Based on a literature study, tests and previous experience, the RANSAC algorithm<sup>15</sup> could be identified as the most suitable for this work.

Nevertheless, when applied to building data, some limitations remain. An example of such limitations of the RANSAC algorithm<sup>16</sup> is presented in Figure 8. The figure reveals that both the *minimum number of support points per primitive* and *maximum distance to primitive* need to be selected carefully. The *maximum distance to primitive* controls the minimum acceptable distance of a point to the plane, *minimum number of support points per primitive* defines the minimum number of points for a plane to be accepted. Depending on the degree of noise and the characteristics of the data, setting the correct input parameters requires human interaction and experience. This can be challenging due to several factors. (i) It is likely that there is no global parameter setting that yields good results for all wall instances. (ii) Setting the parameters correctly requires a good intuition and experience based on a previous visual inspection of the data.

The planes fitted using RANSAC will then be grouped together based on a parallelity constraint and a maximum distance of parallel planes, i.e. the maximum wall thickness expected. The result of these procedures are sets of points per wall instance. Any points outside the wall surface plane, occurring through objects attached to the wall, clutter close to and assigned to the wall cluster, have been cleaned from the data. Thus, the next processing step, namely bounding box fitting, yields more accurate results.

---

<sup>15</sup> Ruwen Schnabel, Roland Wahl, and Reinhard Klein. “Efficient RANSAC for Point-Cloud Shape Detection”. In: *Computer Graphics Forum* 26.2 (2007), pp. 214–226.

<sup>16</sup> Same as above.

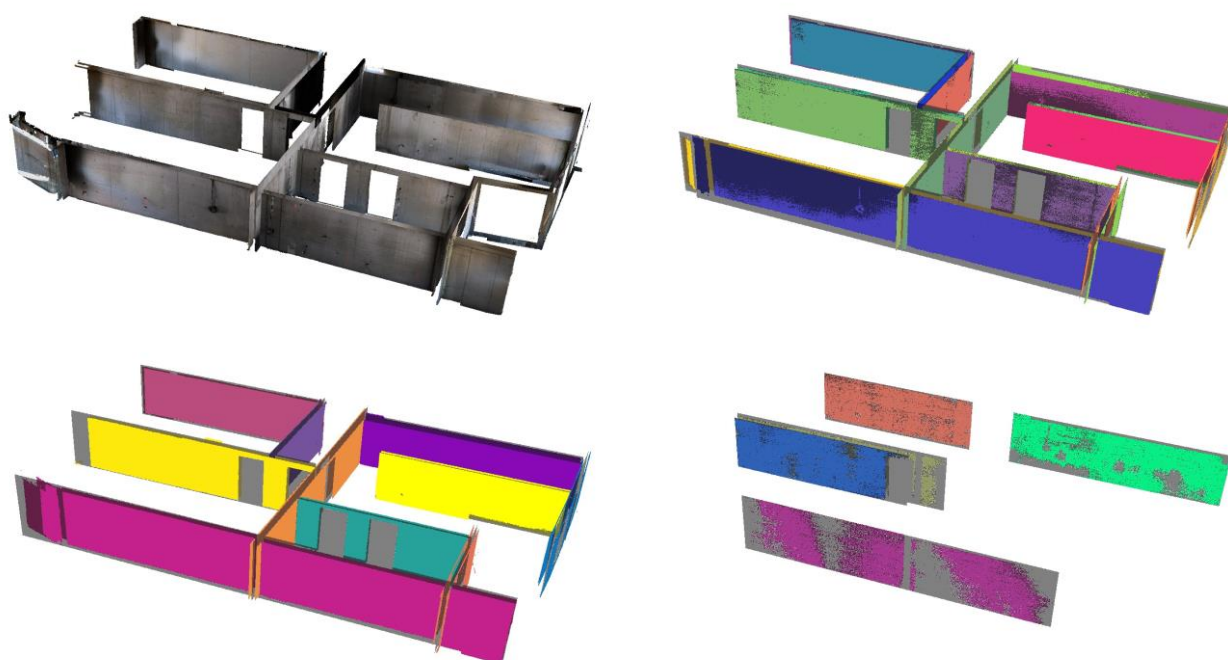


Figure 8 Good and bad results of RANSAC plane fitting on wall points. Top left: original input point cloud. Bottom left: manually segmented. Top right: good result, two planes (one per wall surface) were fitted with minimum number of support points per primitive = 50.000 and maximum distance to primitive = 12 cm. Wall thickness was approx. 20 cm. Bottom right: incomplete segmentation with minimum number of support points per primitive = 50.000 and maximum distance to primitive = 0.1 cm. CloudCompare<sup>17</sup>, which implements the efficient RANSAC as proposed in<sup>18</sup>, was used for the experiments.

#### 3.4.4 Bounding box fitting

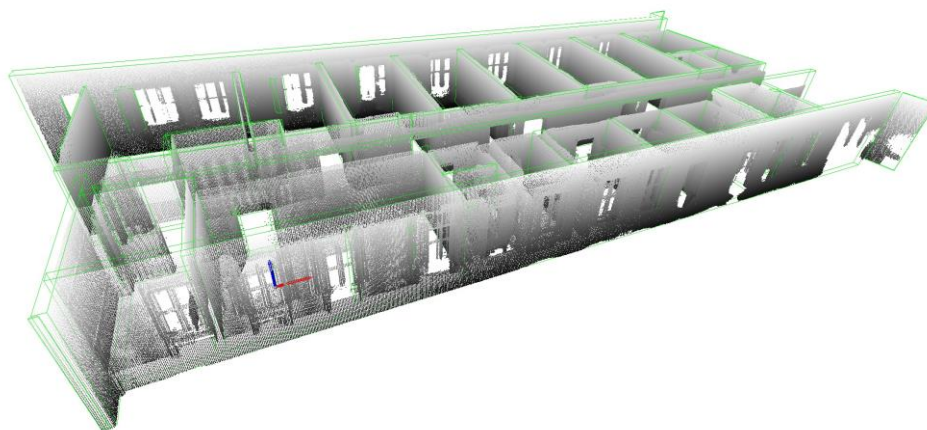
The key process to retrieving the geometric parameters is bounding box fitting. Although a procedure not too complicated, this step only yields realistic geometric parameters when the data is treated as previously described including instance segmentation, plane fitting and grouping which implicitly cleans the data from noise. Directly applied to wall instances, the noise would result in too big dimensions of the bounding box retrieved.

Obviously, an axis-aligned bounding box does not provide too much information about the set of points encapsulated. Neither does it provide an accurate dimension, nor translation or rotation in any direction. However, axis-aligned bounding boxes are easy to fit to the data, as the minima and maxima per coordinate component need to be found. On the other hand, an oriented bounding box provides accurate dimension, translation and rotation of the data.

<sup>17</sup> Daniel Girardeau-Montaut. CloudCompare: 3D point cloud and mesh processing software. 2020. URL: <http://www.cloudcompare.org/> (visited on 10/06/2020).

<sup>18</sup> Ruwen Schnabel, Roland Wahl, and Reinhard Klein. "Efficient RANSAC for Point-Cloud Shape Detection". In: Computer Graphics Forum 26.2 (2007), pp. 214–226.

A suitable algorithm can be found in Open3D. This method fits an axis-aligned bounding box to the data from the frame of each triangle in the 3D convex hull<sup>19</sup>. By estimating the volume of the bounding box fitted, the minimum oriented bounding box can be identified.



*Figure 9 Wall bounding boxes and wall points.*

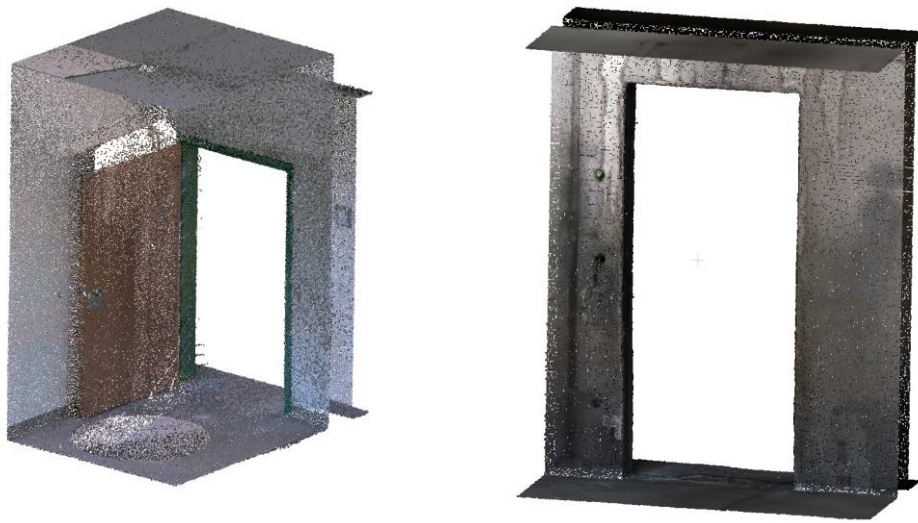
The algorithm for bounding box fitting is meticulously applied to wall segments previously processed by the RANSAC and parallel plane grouping algorithm. This operation is conducted on a per-storey basis. The outcome of this fitting process is a collection of bounding boxes that effectively approximate the geometry of the walls. An example of these wall bounding boxes is presented in Figure 9. It can be observed that the bounding boxes reconstructed represent the data well but still suffer from incorrect topology and unreasonable geometry. The key observations include that separate wall instances that share the same centreline are not connected, gaps occur at wall corners and small bounding boxes are present in the data, which should either be merged with larger walls along the same direction or treated as artefacts. Motivated by these observations, topology-aware bounding box correction routines are implemented.

### 3.5 Doors reconstruction

In BIM, opening objects are represented as child objects of enclosing objects. This is the case for doors, which typically exist as the child object of a wall. The core principle to reconstructing such child objects is to use the parent object's geometry, identify points assigned to the child object and reconstruct the child object. In the following, the method is described for doors only, as it can be adopted to process windows easily.

---

<sup>19</sup> Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. "Open3D: A Modern Library for 3D Data Processing". In: arXiv:1801.09847 (2018).

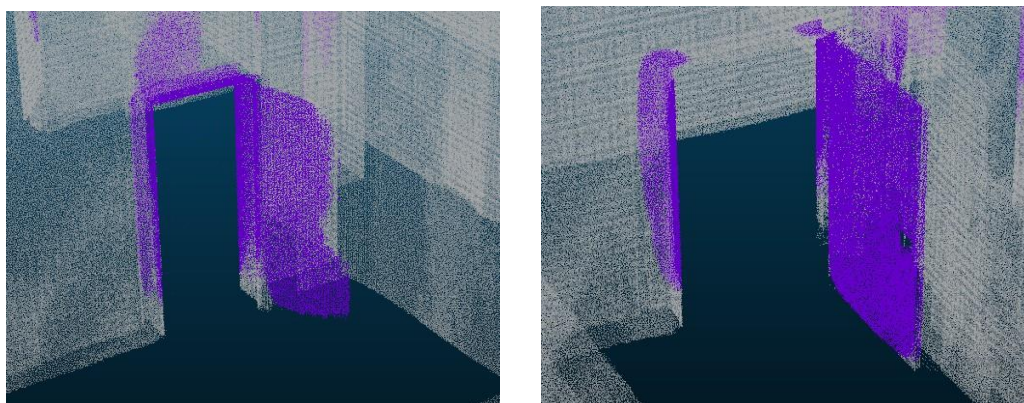


*Figure 10 Sections of scanned scenes. Left: Door with frame, lining and door leaf points that can be segmented. Right: Plain opening, no dedicated objects to be segmented.*

A door incorporates a dedicated object inside an opening comprising a door frame, lining and one or more door leaves, panels, etc. These objects are represented by points in the input data and can thus be segmented and reconstructed. On the other hand, plain openings of walls can fulfil the same purpose as a door, i.e. to connect two spaces, but an opening as such does not contain any dedicated objects and will thus be represented by a void space in the data. Such voids in objects are notoriously hard to reconstruct and require a completely different approach than the one described here. One example of each is given in Figure 10.

The reconstruction is initiated by searching for door points in and close to every potential parent object, in this case in each previously reconstructed wall instance. Any door points assigned to the parent object will then be clustered to separate the individual door instances. After these procedures, an oriented bounding box is fitted to the data, which is projected into the parent geometry to account for flush surfaces of the reconstructed doors and parent walls. In contrast to wall and column reconstruction, the IFC door object needs to be created while iterating all parent objects. This is related to the fact that the parent object needs to be assigned to each door object. The most practical solution is to directly create IFC door objects to avoid the error-prone later assignment of parent IDs.

As doors are child objects of walls, the first step in the process is to find door points associated to the respective wall object. As doors are kept open during data acquisition to ensure a good overlap of



*Figure 11 Examples of segmented doors. Left: good segmentation with door leaf and door frame. Right: under segmentation of door frame and door leaf. Wall points are displayed transparently.*

multiple scan positions or in mobile scanning procedures, a two-step approach is followed to ensure that all door points are assigned to the wall. First, any door points inside the wall bounding box are identified. Naturally, these represent parts of the door frame, lining and door leaf when closed during scanning. Only, if such points are present, the wall bounding box is expanded to find all points of the door instances, mostly covering the door leaves if the door was kept open. By following this procedure, it can be assured that doors are assigned to the correct walls with the opening and door frames. When directly applying the expanded search space, errors might occur as open doors adjacent to perpendicular walls (see Figure 11, right side) might be assigned to the wrong wall. The result of this procedure is that a number of door points are assigned to the wall object. However, these points might still contain several door instances. To find these, DBSCAN<sup>20</sup> is applied, which identifies point clusters forming the door instances.

Per closed door cluster, a bounding box is fitted to the points. Depending on the previous processes, two inconsistencies may occur. On the one hand, the bounding box might exceed the wall width, as door frame, linings and even the door leaf might be on or outside the wall surfaces. On the other hand, under segmentation may lead to an incomplete bounding box reconstruction with the resulting bounding box inside the wall bounding box. In both cases, the door bounding box needs to be projected into the wall bounding box thus ensuring that both the wall and door bounding box vertical intersecting surfaces are flush. Specific procedures are applied to transform the door bounding boxes.

---

<sup>20</sup> Martin Ester et al. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. KDD’96. AAAI Press, 1996, pp. 226–231.

### 3.6 Columns reconstruction

The layout of a building is primarily determined by walls and elements attached to them, like doors. Walls may bear loads, but other structural components like columns, beams and trusses also contribute to supporting the building. Unlike walls, these components do not have sub-elements but they can be interconnected in a way that forms the building's framework, such as the combination of columns and beams in a skeleton structure. Similar to how walls are divided across different floors of a building, these load-bearing components are also typically separated by levels. This discussion will focus specifically on columns to illustrate how such structural elements can be reconstructed.

In alignment with prior reconstruction methods, the process begins with a collection of points classified as 'column' during semantic segmentation. The segmentation of these points into distinct columns is accomplished using DBSCAN<sup>21</sup>. This approach is effective and dependable, as columns are usually spaced apart, facilitating straightforward clustering. Artefactual points in smaller groups are filtered out by imposing a minimum points per cluster limit. This limit is determined based on the point cloud's density, typically ranging from around 100 to 5000 points.

Depending on the shape of the column instance, either bounding box fitting or RANSAC cylinder fitting can be employed to retrieve the geometry of the object.

The bounding box fitting algorithm is detailed in previous chapters. Following a similar method, this process involves projecting all points onto the XY base plane, fitting a convex hull encompassing all data fit a bounding box for each convex hull edge thus minimizing the volume of the bounding box created. It is important to note that in cases of over segmentation, floor and ceiling points adjacent to the column may be included in the cluster, leading to bounding boxes that are excessively large and not accurately representative of the column itself.

---

<sup>21</sup> Martin Ester et al. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. KDD'96. AAAI Press, 1996, pp. 226–231.

Alternatively, parts of the column affected by potential over segmentation can be omitted from the reconstruction by horizontally clipping the column just above the floor and below the ceiling. The remaining section of the column can then be used for a more precise bounding box reconstruction. This is illustrated in Figure 12. On the left side, a horizontal slice is displayed resulting in a reconstructed bounding box with approx. 0.5 x 0.7 m cross section. On the right side, however, the over segmented column instance is shown with a cross section of 1.1 m x 1.1 m.

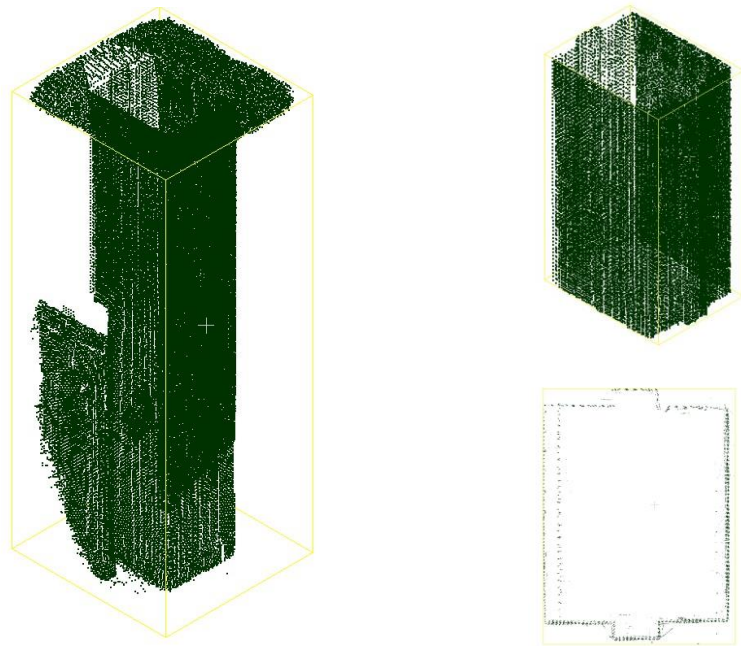


Figure 12 Column over segmentation and horizontal slicing for an accurate reconstruction. Left: segmentation of column instance. Right: horizontal slice providing an accurate reconstruction.

In a manner akin to the RANSAC method previously described, certain constraints can be implemented in the process of fitting a cylinder to a column. A key constraint is the assumption that columns are vertical. By applying this, the cylinder fitting task simplifies to a 2D problem, achievable by projecting all points onto the XY base plane within a Euclidean space. The column's vertical extent is determined by the point set's global minimum and maximum z-values. To fit a circle to the projected points, a random selection of three points is used, and a circle is then fitted to these. With the defined circle and the known minimum and maximum z-values, a cylinder can be constructed. The optimal cylinder fit is determined by iteratively minimizing the distance from the points to the cylinder. Similar to standard RANSAC methodologies, the number of iterations is an adjustable parameter of the function, which ultimately yields the cylinder's centre point, radius, and axis.

## 4 *Pystruct3d*: efficient geometry reconstruction algorithms

In conjunction with this research, we have created and released two open source Python libraries: *openbimxd* and *pystruct3d*. The *openbimxd* package, which includes techniques for creating BIM objects from the reconstructed data, will be explored in the following chapter. Meanwhile, *pystruct3d* is presented in this section, encompassing implementations of all the algorithms discussed in this chapter, complete with an extensive API.

*Pystruct3d* is composed of three distinct modules: (i) The *bbox* module, which includes a class for bounding box representation along with methods for their fitting, manipulation, and combination. (ii)

The *metrics* module, which is equipped with methods to assess the precision of bounding box reconstructions against established ground truths. (iii) The *visualization* module, featuring a visualizer class designed for the display of bounding boxes, individual points, and point clouds, serving as a tool for testing and diagnostic purposes. The *pystruct3d* module leverages high-performance libraries like NumPy<sup>22</sup> and SciPy<sup>23</sup> to optimize computational efficiency. In the realm of point cloud handling, Open3D<sup>24</sup> is utilized, providing a plethora of methods and algorithms applied in this study. The visualizer class also employs Open3D<sup>25</sup>. The package is available at <https://github.com/humantecheu/pystruct3d>. The three modules along with their classes and methods will be described in the following sections.

The subsequent descriptions maintain consistency with the Python module’s nomenclature, with the class and method names written in *italics*.

#### 4.1 ***bbox*: Fitting and manipulating bounding boxes**

The bounding box module comprises the *BBox* class, which represents a bounding box by eight corner points stored in an 8 x 3 NumPy<sup>26</sup> array. During development, it became obvious that a convention on the order of the points in the *BBox* object would be beneficial. This convention is presented in Figure 13.

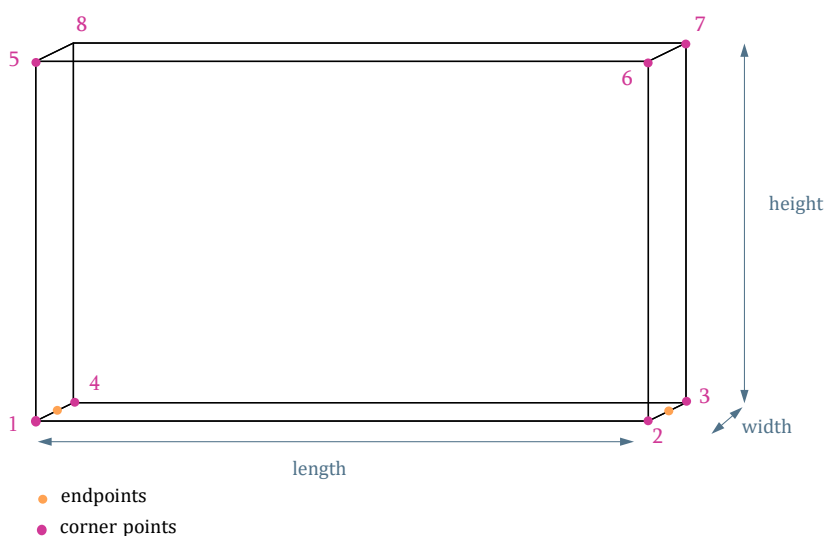


Figure 13 Definition of a bounding box

To ensure a consistent assortment, the bounding box corner points are ordered using the *order\_points* method. More specifically, the points are ordered in a counterclockwise direction from bottom to top. Additionally, two conditions are enforced. (i) The edge connecting the first and

<sup>22</sup> Charles R. Harris et al. “Array programming with NumPy”. In: Nature 585.7825 (2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

<sup>23</sup> Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: Nature Methods 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

<sup>24</sup> Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. “Open3D: A Modern Library for 3D Data Processing”. In: arXiv:1801.09847 (2018).

<sup>25</sup> Same as above

<sup>26</sup> Charles R. Harris et al. “Array programming with NumPy”. In: Nature 585.7825 (2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

second point must be longer than the one connecting the second and third point, i.e. the edge of the first and second point is along the length of the object. (ii) The first point's X and Y coordinate must be smaller than the second point's X and Y coordinate.

Basic methods for transformation include rotation and translation. On a higher level, the *split\_bounding\_box* method can be used to split the bounding box in two. This procedure modifies the box it is applied to and returns another one which represents the other half of the split bounding box. The *axis\_align* method is used to rotate the bounding box along the coordinate frame, whereas the *project\_into\_parent* method is applied to modify a bounding box flush to the vertical surfaces of another one.

To calculate and return basic dimensions of the bounding box, methods including *length*, *width*, *height*, *angle* and *volume* have been implemented. Similarly, other methods calculate and return the endpoints, center plane, and side planes. The *points\_in\_bbox* method can be used to retrieve all points inside a bounding box. This method either applies a convex hull or a particle density function to retrieve the points inside the bounding box from an array of  $N \times 3$  input points.

The *fit...* methods are used to fit either an axis-aligned or horizontally aligned minimum bounding box to a set of input points. As the fit methods are the most crucial for scan-to-BIM pipelines, this field is still under development, seeking more effective, robust, and efficient algorithms. However, the hobb fitting algorithm is implemented and used in this work.

## 4.2 ***visualization: visualize bounding boxes and input data***

The visualization class is essentially a wrapper around some of the Open3D visualization functions and classes. The core idea is to instantiate an object of the visualizer class, add geometry to it and invoke the window with the rendered geometries using the *visualize* method. The methods to add geometry to the visualizer object encompass:

- *point\_cloud\_geometry* to visualize a set of points provides as a NumPy<sup>27</sup> array.
- *o3d\_point\_cloud* to add an Open3D<sup>28</sup> point cloud to the visualizer. Essentially a shorthand of the previous one, as point sets are both stored as NumPy arrays and Open3D point clouds.
- *bbox\_geometry* to render a bounding box as a set of lines representing the box edges

---

<sup>27</sup> Charles R. Harris et al. "Array programming with NumPy". In: Nature 585.7825 (2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

<sup>28</sup> Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. "Open3D: A Modern Library for 3D Data Processing". In: arXiv:1801.09847 (2018).

- *points\_geometry* to render single points as mesh spheres, which is mostly used to visualize geometry corner points. Not suitable for a large number of points.
- The *clear* method can be used to remove all geometry from the visualizer object.

### 4.3 [metrics: evaluating the reconstruction accuracy](#)

In section 6.2 several approaches to calculating the accuracy of the reconstruction will be proposed. Any of these are implemented in the *metrics* module. Applying these and evaluating the accuracy of the reconstruction is part of the pipeline integration in chapter 6.

Likewise, the introduction of the geometric reconstruction algorithms and their implementation, the procedures, tools and implementation used for open source BIM authoring will be introduced in the next chapter.

## 5 Open source BIM authoring

Details on the open source BIM authoring tools tailored for the HumanTech project can be found in D2.2. To ensure a comprehensive documentation here, the core procedure to transform the previously reconstructed geometry into BIM objects will be briefly outlined.

### 5.1 [From bounding box to IFC geometry](#)

BIM representations comprise both geometric and semantic information. For the former, the previously reconstructed geometry (see chapter 4) needs to be transferred into BIM geometric representations which are derived from the geometric reconstruction, e.g. bounding boxes.

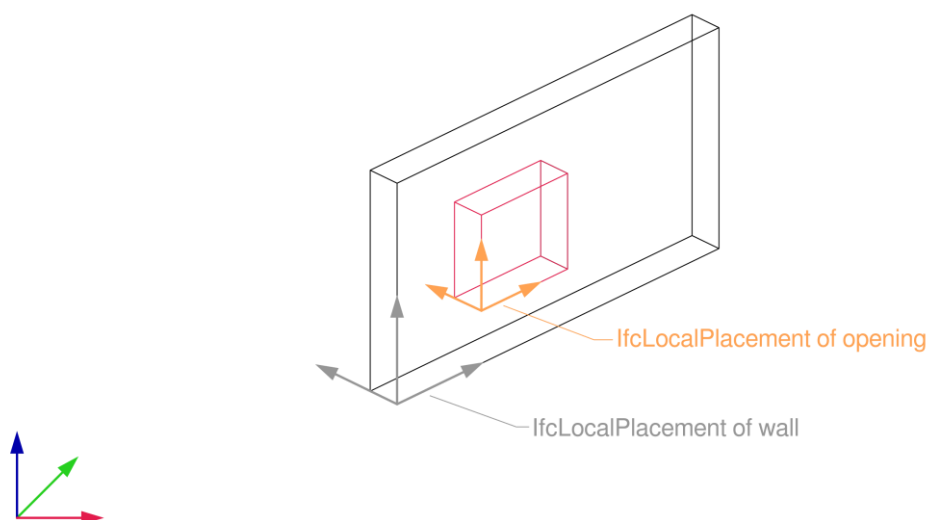
The purpose of this work is not to discuss the overall structure of the IFC standard. However, the basic concept of geometry is vital for further considerations. An IFC element, if the standard allows it to incorporate geometry, will be assigned an *IfcRepresentation*, which is assigned an *IfcShapeRepresentation*. Among basic geometries such as points or curves, an *IfcShapeRepresentation* can include parametric solid geometries such as Swept solid, Boundary representation and Constructive solid geometry<sup>29</sup>. All mentioned concepts have particular advantages. However, for cuboid-like geometry as reconstructed previously, the *IfcSweptSolid* is the primary choice. The basic principle is to use a profile and extrude it along a curve. Thus, to create a box as an *IfcSweptSolid*, a rectangle is used as a profile, which is then extruded along the normal of the rectangle with a given dimension.

Technically, such parametric IFC geometry could be created using global coordinates, i.e. the corner points of the bounding boxes previously reconstructed. However, IFC models follow the concept of local placements. A local placement is a local coordinate frame defined by its axes as vectors and the

---

<sup>29</sup> BuildingSMART. Industry Foundation Classes: Version 4.2 bSI Draft Standard IFC Bridge proposed extension. Ed. by BuildingSMART. 2019. URL: [https://standards.buildingsmart.org/IFC/DEV/IFC4\\_2/FINAL/HTML/](https://standards.buildingsmart.org/IFC/DEV/IFC4_2/FINAL/HTML/) (visited on 10/01/2022).

coordinates of the origin. The concept of local placement applies to the overall structure of the IFC model, including `IfcProject`, `IfcSite`, `IfcBuilding` and finally, the building components. According to the standard, all `IfcElements` (`IfcWall`, `IfcColumn`, ...) should be placed relative to the local placement of its container, e.g. `IfcBuilding`<sup>30</sup>.



*Figure 14 IfcLocalPlacement of IfcWall and IfcOpeningElement*

The same principle applies to child elements such as openings that are associated with a wall, as illustrated in Figure 14. The origin of the `IfcLocalPlacement` is defined with respect to the parent `IfcLocalPlacement` as defined in the IFC documentation. Mainly, the length of the object should expand in the x direction of the `IfcLocalPlacement`, and the main edges of a cuboid-style geometry should be aligned with the coordinate frame axes, with Z as the upright. The origin of the `IfcLocalPlacement` should be set at the 'lowest-left corner' depending on the rotation. The length of the object, however, should expand in x-direction.

The described concept requires the global bounding box coordinates to be transformed into a local coordinate frame, or a local coordinate frame relative to the parent element, respectively.

As the points in the bounding box are presented in a specific order, the `IfcLocalPlacement` origin can be set to the first bounding box point, the X axis is the norm of the vector connecting the first and second point, the Y axis can be set as the vector connecting the first and fourth point, the Z-axis is set vertical. To ensure that all axes are perpendicular, only the origin and rotation are passed to `IfcOpenShell's` <sup>31</sup> `util.placement.rotation` function.

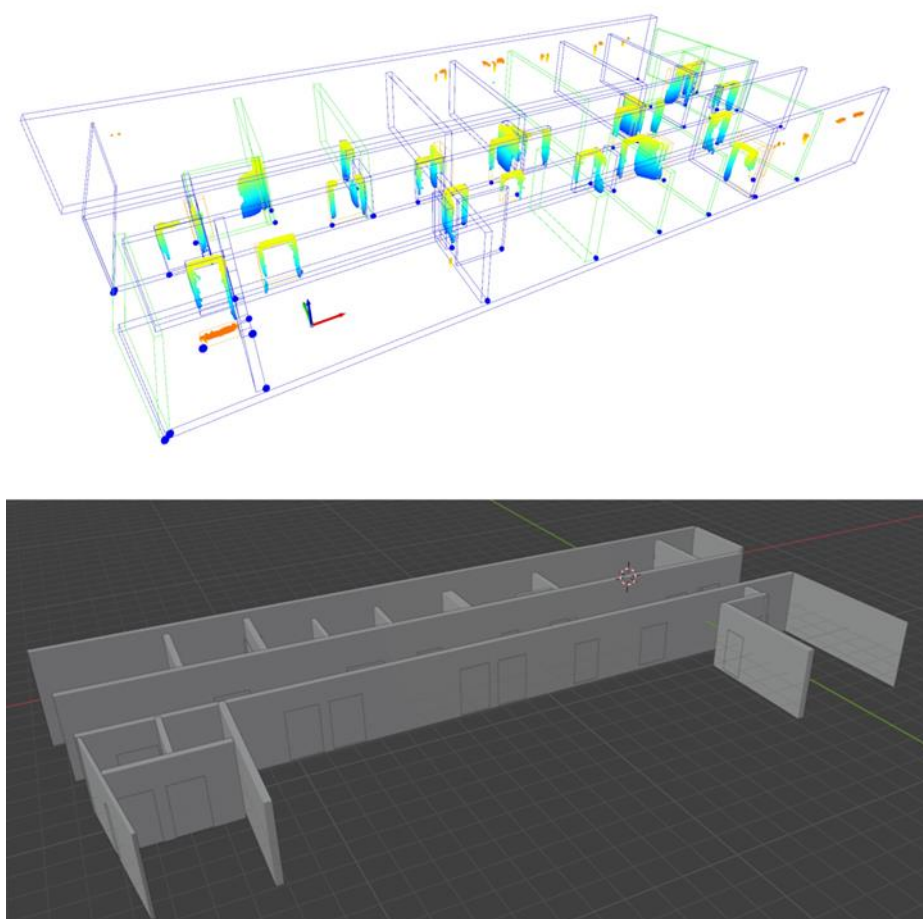
<sup>30</sup> Same as above

<sup>31</sup> Thomas Krijnen. `IfcOpenShell`. 2021. URL: <http://ifcopenshell.org/>.

For the `IfcLocalPlacement` of any opening, a copy of the wall local placement that has been transformed to the opening position can be used. As the door has been projected into the wall bounding box, the transformation vector is the one connecting the first points of the wall and door bounding boxes. The points of both bounding boxes need to be ordered before.

The described concept and procedure can be used for any object with cuboid geometry (walls, slabs, square columns, ...) and respective child elements (doors, shafts, openings, ...). The `IfcLocalPlacement` of round objects, e.g. round columns is placed at the centre of the lower circle and the orientation is aligned with the global X and Y axes, the Z axis with the vertical column centre axis.

To demonstrate the behaviour, Figure 15 shows wall and door bounding boxes with the IFC geometry. The blue points represent the origin of the `IfcLocalPlacement`, which aligns with the first corner point of the bounding boxes. In the lower part of the figure, the resulting geometry solids are displayed. Note that some elements have been hidden to allow for a better view.



*Figure 15 Wall and door IFC elements. Top: door geometry with door points. Points mark the origin of the IFC local placement. Bottom: Resulting IFC objects.*

## 5.2 Semantic IFC object creation

In the context of this work, three types of IFC concepts for the enrichment of models with semantic information apply: *IfcMaterial* creation and assignment, property sets and spatial relationships. Note that the scope of this work is limited to visual features and mainly the geometric reconstruction. Thus, any information on material and other properties are either captured visually or set as generic values to demonstrate the assignment of both material information and other properties.

### 5.2.1 Material information

In the course of the IFC model creation, the *IfcMaterial* definitions can be set up as one of the first steps, as these can be used for all objects created. Besides the plain *IfcMaterials* creation, other concepts such as *IfcMaterialLayerSets* apply for elements such as walls. The *IfcMaterialLayerSet* entity consists of a set of layers with specified thickness and assigned material. Although an *IfcMaterialLayerSet* would be the right choice for elements such as walls, in this work we only assign an *IfcMaterial* to the elements reconstructed, as the layer composition is unknown.

### 5.2.2 Property sets

Beyond material assignment, the concept of property sets is relevant to most building objects in the IFC schema. These property sets act as templates for assigning semantic information to specific elements. Specifically, the standard property set for walls, denoted as *PSet\_WallCommon*, includes properties like *Reference*, *AcousticRating*, *FireRating*, *Combustible*, *SurfaceSpreadOfFlame*, *ThermalTransmittance*, *IsExternal*, *ExtendToStructure*, *LoadBearing*, *Compartmentation* and *Status*<sup>32</sup>. Even without detailed explanations, it is evident that these properties are intended for the standard occurrences of walls, aligning with the primary objective of property sets. In addition to the standard ones, specific property sets can be created to meet the particular requirements of projects and organisations.

### 5.2.3 Spatial relationships

In IFC, objects are assigned to spatial containers. In buildings, storeys and levels are the common type of spatial containers. The level elevations are one of the first information inferred in the proposed pipeline, and can be used to set the elevation of *IfcBuildingStorey* objects. Any object of the storey is then assigned to the respective spatial container using an *IfcRelContainedInSpatialStructure* object. Note that the information on the spatial relationship of an object is not contained within the object itself, but in a separate class object.

---

<sup>32</sup> Names are written as in the IFC schema, ignoring correct spelling

## 6 Experiments and results

Experiments were performed on the 14NH dataset. The experiments will clearly reveal the potential of the proposed framework.

Results are reported based on two common metrics including Intersection over Union (IoU) and the centroid deviation. The former is calculated by dividing the volume of the intersection of both the reconstructed and reference geometry by the union of both. The latter measures the distance of both centroids.

Beyond the mere results, an ablation study examining the advances of the proposed reconstruction pipeline and assessing the effect of the semantic segmentation accuracy will be presented. One of the scenes of the 14NH dataset is used for the ablation study.

### 6.1 CV4AEC challenge

The interdisciplinary HumanTech team submitted 2023 and 2024 to the Computer Vision in the built environment (CV4AEC) 3D reconstruction challenge. The team could achieve the third place in both years (<https://cv4aec.github.io/#winners>). The challenge offers training and testing point clouds from various types of buildings. The training dataset was augmented with manual annotations and used to deliver the segmentation model (see D3.5) and test the Scan to BIMxD pipeline.

Our best results on the 7 CV4AEC test scene submitted 2024 are reported in Table 1. One reconstructed BIM is shown in Figure 16. The results were calculated using the evaluation server provided by the challenge organizers. Note, that each scene marked with \* combines the results of two floors of the same building processed independently. The *08\_ShortOffice* provides the best average result in all three categories. Although *34\_Parking's* average IoU is slightly higher, no columns are considered.

Metrics	08_ShortOffice_01*	11_MedOffice_05*	25_Parking_01*	34_Parking_04	Overall
Wall bounding box IoU [-]	0.217	0.220	0.262	0.307	0.243
Door bounding box IoU [-]	0.292	0.138	0.034	0.278	0.172
Column bounding box IoU [-]	0.363	-	0.351	-	0.357
Average bounding box IoU [-]	0.291	0.179	0.216	0.292	0.241

*Table 1 Results of the CV4AEC challenge. The results marked with \* are the average of two floors of the respective building.*

Considerably interesting about the CV4AEC testing scenes is the fact that various types of buildings are encompassed. One being an office building with individual rooms, one being an open-plan office building with cubicals, one being a parking lot and the last one comprising a scan of a parking lot entrance building. The pipeline thus faces a wide range of features and objects to reconstruct.

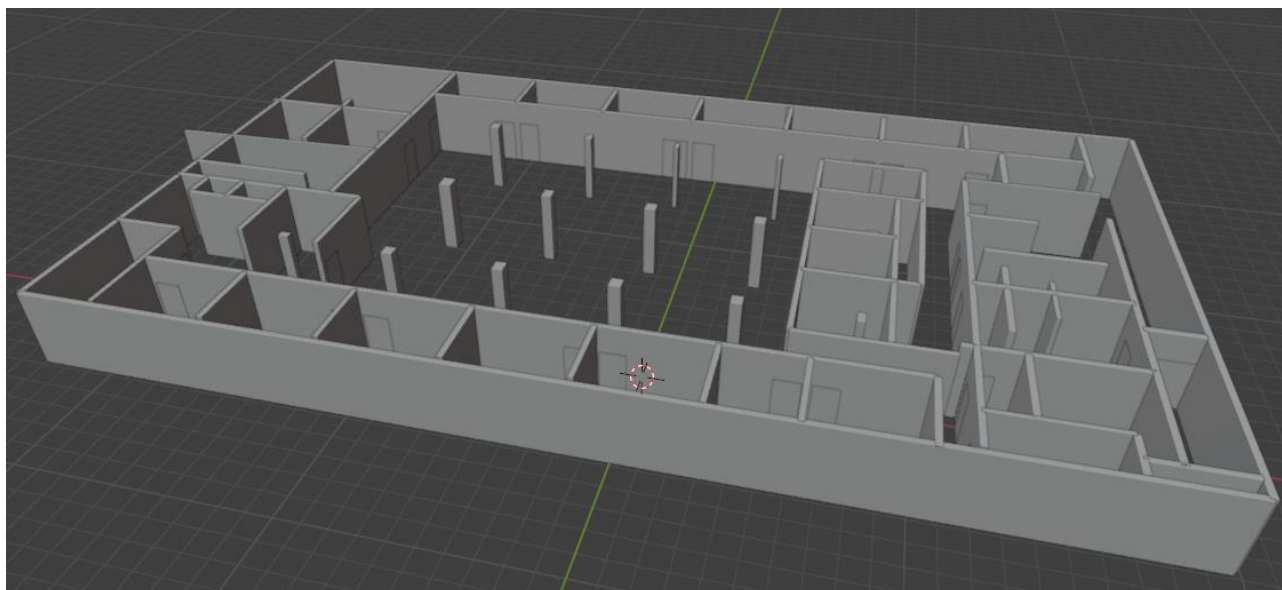


Figure 16 Reconstructed BIM of the 08\_ShortOffice\_01\_F1 scene.

## 6.2 14 Nothelfer hospital, Weingarten, Germany

The 14NH dataset comprises four scenes from three different buildings. The first two scenes were captured in Building A, which features office rooms on the first floor and patient rooms on the second floor. This building, approximately 100 years old, has a classic layout with structural walls, a central hallway, and small terraces on the second floor. Both floors in Building A were largely unfurnished. Building B includes an area from a former Intensive Care Unit (ICU), where the captured rooms were mostly empty except for integrated furniture and clinical equipment, such as terminals with power and gas outlets mounted on movable arms attached to the ceiling. Compared to Building A, the doors and hallways in Building B are wider, meeting the requirements for an ICU section in a hospital. Finally, scenes from surgery rooms in Building C were captured. Similar to the ICU in Building B, the surgery rooms were empty except for gas and power outlets on movable arms and integrated furniture. The doors and hallways in Building C are similarly wide. For Building A, an aligned scan of the building's exterior is available and was used for processing; for the other buildings, only the inner surface of the exterior walls is included in the dataset.

All four scenes were reconstructed using the presented pipeline. The results are presented in Table 2. No columns were present in the A first floor and C surgery scene, thus '-' was reported for the column metrics.

In the second-floor scene of Building A, a notable decrease in both the walls' and doors' IoU and vIoU compared to the first floor is observed. The primary issue stems from the wall configuration, which forms small compartments, likely used as lavatories, on the backside of the hallway walls. A visual comparison indicates that the pipeline struggles to accurately reconstruct these small wall segments.

This difficulty is linked to the wall merging and closing procedures (refer to section 3.5), which tend to merge these small segments into larger wall instances.

Metrics	A 1st floor	A 2nd floor	B ICU	C surgery
Wall centroid deviation [m]	0.334	0.309	0.183	0.156
Wall bounding box IoU [-]	0.395	0.219	0.270	0.328
Door centroid deviation [m]	0.158	0.355	0.214	0.184
Door bounding box IoU [-]	0.417	0.104	0.193	0.223
Column centroid deviation [m]	-	0.191	0.200	-
Column bounding box IoU [-]	-	0.037	0.095	-

Table 2 Results of the 14NH reconstruction

Beyond this result overview, an ablation study can support the understanding of benefits that specific methods yield. Thus, three different configurations were examined on the A building first floor scene of the 14NH dataset.

- The **baseline** method includes the full reconstruction pipeline starting with the semantic segmentation, but does neither apply any topology refinement including height correction, wall merging and closing and door splitting, nor any dimension threshold or correction. Instead of HYSAC, RANSAC is used for primitive fitting.
- The **scan-to-BIMxD pipeline** method includes all algorithms and procedures as described in the previous sections and encompassing semantic segmentation, topology correction and HYSAC plane fitting.
- To assess the effect of the semantic segmentation quality on the overall process, **manual labels** are used instead of the labels inferred by the transformer-based segmentation model.

Metrics	baseline	Scan-to-BIMxD pipeline	manual labels
Wall centroid deviation [m]	0.402	0.334	<b>0.330</b>
Wall bounding box IoU [-]	0.371	<b>0.395</b>	0.346
Door centroid deviation [m]	0.218	<b>0.158</b>	0.173
Door bounding box IoU [-]	0.322	0.417	<b>0.519</b>

Table 3 14NH ablation study.

Considering the numbers presented in Table 3 it remains to be noted, that the *Scan-to-BIMxD pipeline* approach yields considerable advantage regarding both wall and door reconstruction accuracy. This demonstrates the potential of the *Scan-to-BIMxD pipeline* approach. However, this particular dataset has very low occlusions and no interior objects. Especially the wall segmentation is very accurate. Remarkably, the manual labels do not outperform the wall reconstruction of the *Scan-to-BIMxD pipeline* approach, that yields a 4.9% higher bounding box IoU.

To further investigate the impact of the quality of the semantic segmentation on the overall reconstruction, a test was carried out using the manual annotations. This reveals, that there is some positive effect on the wall reconstruction accuracy, but a major positive effect could be observed for the reconstruction quality of doors, with the door IoU increased by 10.2%. The major conclusion is that

although the *Scan-to-BIMxD pipeline* approach can cope with incomplete segmentation, as observed for doors in this dataset, a better segmentation accuracy boosts the results.

### 6.3 KPIs

The main metric reported so far is the bounding box IoU. During the course of the project it became obvious that the IoU became a common metric for scan to BIM evaluation. Thus, most of the previous results reported uses this metric. Nevertheless, there are arguments to use other metrics. On the one hand, the IoU does not behave linear and penalizes even small misalignments. On the other hand, calculating the IoU requires a reference BIM, mostly created manually. Although this is common practice and such models are available, a bias induced by the manual modelling process cannot be avoided. To overcome this limitation, the cloud-to-BIM distance can be calculated as a metric independent of a ground truth BIM as it is only calculated using the reconstructed BIM and the input point cloud.

The cloud-to-BIM distance (K3.10 BIM reconstruction quality) was calculated for the 14NH hospital scenes and is reported in Table 4. The distances were calculated as absolute values and only distances of 0.20 m and below were used. Although the Target value of 0.04m could not be reached, the target value for month 18 of 0.08m could be reached.

Metrics	A 1st floor	A 2nd floor	B ICU	C surgery	Average
Cloud-to-BIM distance [m]	0.061	0.063	0.060	0.051	0.059

Table 4 14NH cloud-to-BIM distance

In the process of transferring the Scan-to-BIMxD pipeline into WP7 and the pilot applications, it can be expected that the KPI metric will still improve with the potential to reach the target value of 0.04m at the end of the project.

## 7 Discussion and conclusions

Based on the results provided, it can be concluded that the Scan-to-BIMxD pipeline is a robust approach to deliver BIMs from buildings based on point clouds. One of the main limitations seems to be that only three object classes including wall, door and column are reconstructed in the current implementation. Obviously, this is not sufficient to deliver functional BIMs. In this context, we argue that the algorithms introduced can be easily applied to other elements, which in turn is a matter of implementation and not subject to further research.

Bassier et al.<sup>33</sup> report that the accuracy of the automatically created walls even outperforms the manual modelling. Hereby, the accuracy is evaluated by the minimum Euclidean distance of reconstructed IfcWallStandardCase and the ground truth mesh. Our experiments underpin this finding, as a

<sup>33</sup> Maarten Bassier and Maarten Vergauwen. "Unsupervised reconstruction of Building Information Modeling wall objects from point cloud data". In: Automation in Construction 120 (2020), p. 103338. ISSN: 09265805. DOI: 10.1016/j.autcon.2020.103338.

comparison of the cloud-to-BIM distance of the ground truth BIM and the reconstructed BIM model reveals. For the 14NH A-Building 1st floor, the mean signed distance to the reconstructed BIM was 0.020m, but for the ground truth BIM the mean signed distance is 0.052m. The manually classified wall points and the reconstructed IfcWall objects were used to calculate the distance.

Putting the average cloud-to-BIM signed distance of 0.02m in context to the wall vIoU of 0.553 results in two conclusions. On the one hand, the pipeline approximates the input point cloud accurately with only a minimum distance of the reconstructed objects and the point cloud. On the other hand, the human-created ground truth BIM model contains all the inference and intelligence of the human modeller, who is capable of visually interpreting the configurations of the objects and thus applies corrections that the pipeline proposed in this work does not yet incorporate.

Regarding data availability, two key aspects should be noted. First, the data annotation guidelines, procedures, and datasets significantly influence semantic segmentation accuracy. However, only a fine-tuned model trained on the specific features of the data can achieve the level of accuracy required for the framework proposed here. Consequently, a fine-tuning would have to be performed before processing any data with features differing from the dataset incorporated into the training. Second, despite not being a primary research focus, the availability of training data with high-quality semantic annotations remains a challenge. Initiating a dataset crowd-sourcing effort, similar to the ScanNet dataset<sup>34</sup>, or deriving data from a commercially available service, could be valuable approaches, especially in cases where corresponding BIMs are available to enhance the efficiency of the data annotation process.

Theoretically, achieving 100% accurate semantic segmentation should result in 100% reconstruction accuracy. However, the ablation study revealed that this is not the case, with a maximum wall IoU of only 0.395 being obtained with manually annotated data. This finding leads to the major conclusion of this study: enhanced reasoning algorithms are necessary to infer more geometric information about the observed objects. It is likely that point clouds, even with associated colour values as processed here, may be insufficient for this task, necessitating the incorporation of additional data modalities. On an initial level, instance segmentation and image-based segmentation of sub-elements could be explored. Ultimately, a comprehensive BIM generation model could be envisioned, capable of processing various data modalities — including 2D design drawings as pixels or vectors, images, text, and 3D data — and directly inferring a BIM model in the IFC format.

---

<sup>34</sup> Angela Dai et al. “ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes”. In: (2017). URL: <http://arxiv.org/pdf/1702.04405v2> (visited on 01/28/2022).