

## BIM RECONSTRUCTION: AUTOMATED PROCEDURAL MODELING FROM POINT CLOUD DATA

M. Bassier\*, L. Mattheuwsen and M. Vergauwen

Dept. of Civil Engineering, TC Construction - Geomatics  
KU Leuven - Faculty of Engineering Technology  
Ghent, Belgium  
(maarten.bassier,lukas.mattheuwsen,maarten.vergauwen)@kuleuven.be

### Commission II

**KEY WORDS:** Building Information Modeling, Reconstruction, Walls, Building, Point Clouds

### ABSTRACT:

The reconstruction of Building Information Modeling objects for as-built modeling is currently the subject of ongoing research. A popular method is to extract structure information from point cloud data to create a set of parametric objects. This requires the interpretation of the point cloud data which currently is a manual and labor intensive procedure. Automated processes have to cope with excessive occlusions and clutter in the data sets. To create an as-built BIM, it is vital to reconstruct the building's structure i.e. wall geometry prior to the reconstruction of other objects.

In this work, a novel method is presented to automatically reconstruct as-built BIM for generic buildings. We presented an unsupervised method that procedurally models the geometry of the walls based on point cloud data. A bottom-up process is defined where consecutively higher level information is extracted from the point cloud data using pre-trained machine learning models. Prior to the reconstruction, the data is segmented, classified and clustered to retrieve all the available observations of the walls. The resulting geometry is processed by the reconstruction algorithm. First, the necessary information is extracted from the observations for the creation of parametric solid objects. Subsequently, the final walls are created by updating their topology. The method is tested on a variety of scenes and shows promising results to reliably and accurately create as-built models. The accuracy of the generated geometry is similar to the precision of expert modelers. A key advantage is that the algorithm creates Revit and Rhino native objects which makes the geometry directly applicable to a wide range of applications.

### 1. INTRODUCTION

The production of as-built BIM models has become widespread in the AEC industry. These models are vital in project preparations for renovations and are becoming increasingly mandatory in project deliveries (Kavanaugh, 2013). Additionally, it can be used for managing heritage assets and numerous analyses for existing buildings (Volk et al., 2014, Patraucean et al., 2015). In the case of new constructions, the geometry of an as-built model can be obtained from updating an as-design model. More commonly, it is conceived from metric measurements such as point cloud data of the site. Currently, this Scan-to-BIM process is a manual procedure performed by expert modelers that interpret the point cloud and manually design all the relevant objects in the scene. This is a significant time investment and thus the uptake of automating this procedure is enormous. However, automated procedures are halted by the lack of consistent data, the computational effort of the procedures involved and the complexity of built structures. Furthermore, point cloud data acquired by remote sensing techniques are inherently occluded and are littered with clutter. It is within the scope of this research to fully automatically reconstruct BIM wall geometry up to as-built conditions under these realistic conditions.

Automated Scan-to-BIM processes can be considered either bottom-up or top-down procedures (Hichri et al., 2013). The former considers a range of measurements from the site as input and extracts increasingly higher level information from the observations until the intended geometry can be reliably constructed. This is a general approach which relies on building logic to interpret the data.

It is often used in projects that do not have access to other reliable sources of information such is the case with insufficiently documented buildings and heritage structures. In contrast, the latter relies on this prior information for the point cloud interpretation and reconstructs a well defined number of objects in a selective manner. This is considered a supervised pattern recognition approach which is commonly used in Scan-vs-BIM (Bosché et al., 2013, Bosché et al., 2014). In this research, we propose a bottom-up method since the majority of built structures do not have reliable plan information (Gimenez et al., 2015). Furthermore, we solely rely on the point cloud and building logic for the point cloud interpretation since the incorporation of e.g. sensor information would make the procedure sensor-dependent.

The emphasis of this work is on the reconstruction of wall geometry as it forms the basis of other objects in the Building Information Model. Also, their observed geometry has a quite distinct signature which can be reliably detected by heuristics and machine learning techniques. The presented process consists of a series of consecutive procedures including segmentation, classification, clustering and finally the reconstruction of the objects. This research solely discusses the final step as the first three steps are performed in prior work (Bassier et al., 2018, Bassier and Vergauwen, 2019b). The procedure is specifically designed to deal with the significant amount of clutter in the scene and to reason about the occlusions in the input point cloud. The goal is to compute a set of abstract but accurate LOD200 (BIMForum, 2016) walls based on clustered wall segments. Also, our approach creates both Revit and Rhino native geometry which ensures data compatibility with a wide range of industry applications.

The remainder of this work is structured as follows. The back-

\*Corresponding author

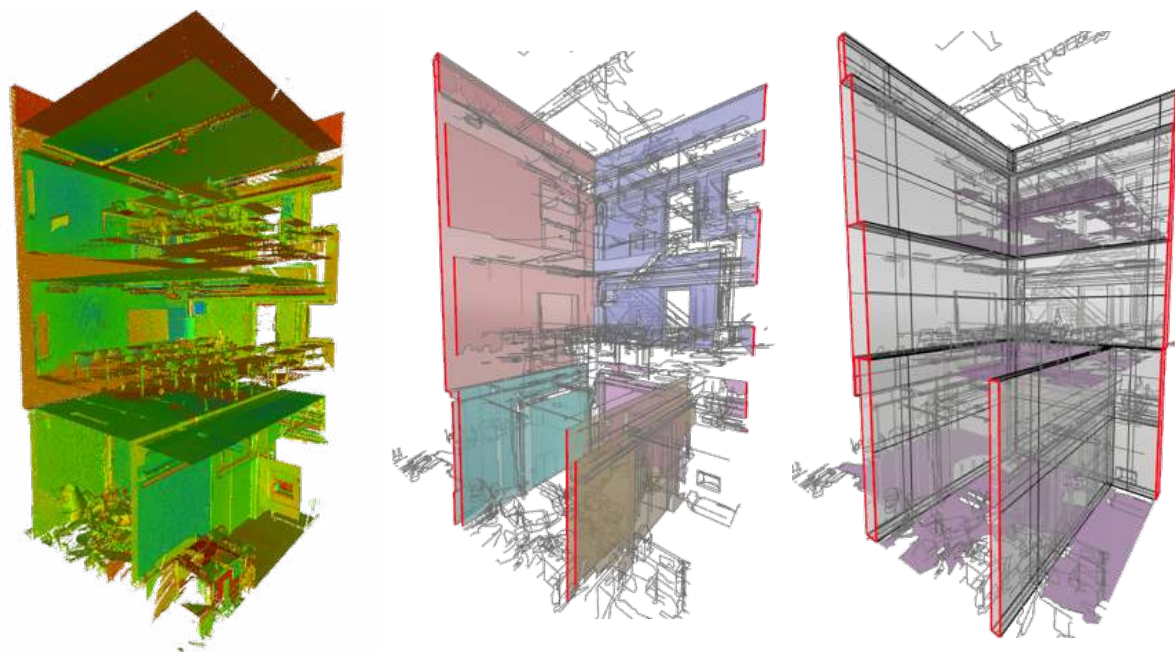


Figure 1: Overview of the intermediate results of the presented Scan-to-BIM procedure: unstructured point cloud (left), extracted classified clustered planar wall segments (mid) and the reconstructed LOD200 walls (right).

ground and related work is presented in Section 2. In Section 3, the methodology is presented. The test design and experimental results are proposed in Section 4. Finally, the conclusions are presented in Section 5.

## 2. BACKGROUND & RELATED WORK

As stated above, bottom-up processes gradually extract higher level information from the point cloud prior to the reconstruction (Nguyen and Le, 2013). Typically, the data is not processed in its entirety but subsampled for efficiency (Tang et al., 2010). In 2D methods, the point cloud is often reduced to a set of raster images consisting of a slice of the data or other information (Landrieu et al., 2017, Anagnostopoulos et al., 2016). In 3D methods, the point cloud is typically subsampled and restructured as a voxel octree which allows efficient neighborhood searches (Vo et al., 2015) (Fig. 1left). Next, the data is interpreted and assigned to one of several predefined classes such as walls, ceilings and floors that form the building's main structure. Single point classification does have applications (Qi et al., 2016) but typically the data is oversegmented into geometric primitives such as planes or cylinders to increase the distinctiveness of the observations. In our research, we segment the data according to planar primitives due to the structural components of a building generally being planar. From each segment, local and contextual information is extracted and processed by reasoning frameworks to assign the appropriate label (Garstka and Peters, 2016, Weinmann et al., 2017, Dittrich et al., 2017). Both heuristics and machine learning techniques are considered, both of which show promising results due to the distinct signature of archetypical building classes (Bassier et al., 2016, Wolf et al., 2015, Xiong et al., 2013, Nikoohemat et al., 2017). Once the class of the segments is established, the data is grouped into all the relevant observations of individual objects. In previous work, we employ a pretrained Conditional Random Field to associatively group planar wall segments into wall clusters (Fig. 1mid). These groups of observations are the input of the presented reconstruction algorithm that extracts the relevant wall parameters and subsequently the topology analysis.

Wall reconstruction algorithms vary depending on the wall geometry that is reconstructed. Most approaches consider wall reconstruction within the scope of surface based applications such as in navigation, robotics, and so on (Díaz-Vilariño et al., 2015). For instance, Xiong et al. (Xiong et al., 2013) and Adan et al. (Adan and Huber, 2011) reconstruct planar wall boundaries and openings based on machine learning. Michailidis et al. (Michailidis and Pajarola, 2016) reconstruct severely occluded wall surfaces using a Bayesian graph-cut optimization on a cell complex decomposition. Even Neural Networks are employed to compute a realistic representation of wall surfaces (Barazzetti, 2018). In this light, it can also be considered as a room reconstruction paradigm. Oesau et al. (Oesau et al., 2014) consider the creation of watertight rooms as a 2D graph-cut optimization problem. Ochmann et al. (Ochmann et al., 2016) and Mura et al. (Mura et al., 2016, Mura et al., 2014) both focus on finding the optimal room layout. Ochmann et al. do fit wall objects on the rooms edges but do this for the purpose of room separation.

The emphasis of this research is on the reconstruction of Building Information Modeling geometry which is most commonly based on solid parametric entities that can be extended with non-metric information for a variety of applications (Volk et al., 2014). This requires a fundamentally different approach since we do not reconstruct the partially observed surfaces of the walls but rather use the observations to extract the parametric information required to create a solid wall object. In this research we create LOD200 walls, which have an abstract physical appearance and require a base constraint, a height or a top constraint, a walltype and a path which serves as a centreline or a centreplane. Closely aligned with our work is the method of Macher et al. (Macher et al., 2017). They have similar outputs as our method but first create .obj files within Matlab which they manually transfer to .ifc files using the FreeCAD Open Source software. In our work, we use the Rhinocommon API for the information extraction and use Rhino.Inside (Robert McNeel & Associates, 2019) and Revit API to fully automatically create BIM geometry directly within Revit. This allows for a wide range of outputs without the sometimes erroneous conversion through .ifc formats and exploits the capabilities of both API's. Thomson et al. (Thomson and Boehm, 2015)

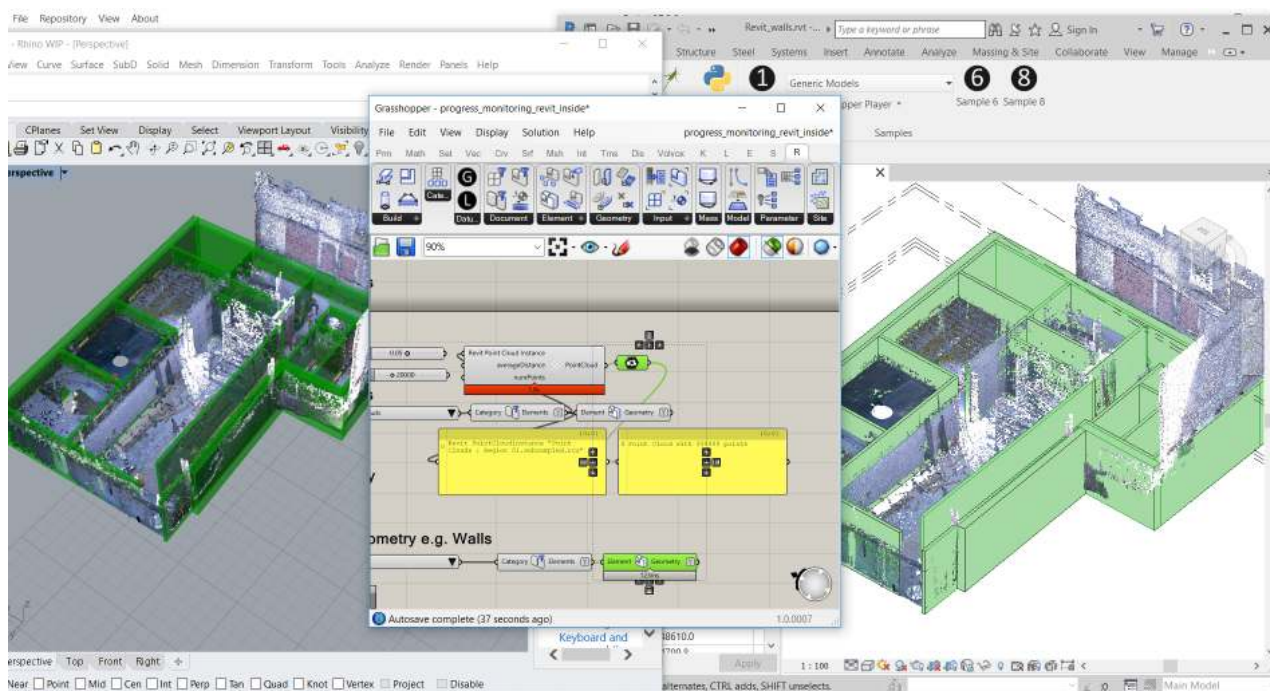


Figure 2: Overview implementation of the reconstruction algorithm. (1) the parameter extraction is performed in Rhino and Grasshopper with developed components based on the Rhinocommon API and (2) the partial wall geometry and final wall geometry are constructed directly inside Revit through the Rhino.Inside and Revit API.

also create BIM geometry. They propose a two-step approach where they first create partial wall geometry of each segment or paired segment. In the second step, they merge coplanar partial walls and heuristically filter outliers. In our method, the clustering of wall segments is a separate step prior to the reconstruction. This allows for a more balanced parameter extraction since all the observations are already available per wall.

The reconstruction of the topology between walls has also been researched. For instance, Budroni et al. (Budroni and Böhm, 2010) and Previtali et al. (Previtali et al., 2014) use wall line intersections in a 2D cell decomposition. They combine both ceiling and floor geometry to create initial blue prints for the rooms. Valero et al. solves the intersections of pre-segmented wall lines to create a closed area (Valero et al., 2012). 3D approaches also have been presented. For instance, Turner et al. proposed 3D voxel carving to create watertight meshes of rooms (Turner and Zakhor, 2014). They determine individual room labels by performing a min-cut on a 2D graph of the Delaunay mesh of the floor plan. They are one of the few researchers that perform a multi-storey reconstruction which also is the goal of this research. However, their emphasis is on room boundaries while this research focuses on accurate wall reconstruction. In this research, the wall intersections are computed using the semi-automated tools within Revit. This is quite efficient and often the connections between objects are project or application dependent.

### 3. METHODOLOGY

In this paper, a wall reconstruction algorithm is proposed that creates BIM geometry directly within Revit. The procedure takes as input a set of clustered labeled wall observations and outputs generic LOD200 Basic walls (BIMForum, 2016) which are compatible with the IFCWallStandardcase class (BuildingSMART International Ltd, 2013). Two steps are defined in the process. First, parametric information is extracted from the clustered labeled segments to assign the appropriate walltype and constraints.

Next, the best fit centreline is computed and the partial geometry is created. The method creates Revit native geometry through the Rhino.Inside Plug-in and combines the Revit and Rhinocommon API. This allows for a flexible method that can serve a wide range of applications and also gives access to multiple file formats. The consecutive steps are discussed in detail in the following paragraphs.

#### 3.1 Data preprocessing

Prior to the reconstruction, the data is segmented, classified and clustered. First, the unstructured point cloud is represented as a voxel octree after which planar patches are extracted from the data as presented in our previous research (Bassier et al., 2017). Next, the planar patches are subjected to a reasoning framework that computes class labels for each patch. A pre-trained Random Forests model is used for the classification (Bassier et al., 2018). The result is a set of labeled segments that replaces the point cloud representation of the building. These are then clustered into groups that represent the individual walls. A Conditional Random Field exploiting local and contextual information is employed to compute the most likely assignment of the wall segments (Bassier and Vergauwen, 2019c). The result is a set of clustered wall mesh segments that represent all the available observations of each wall. This is a highly reliable and accurate representation of the observed structure but it is heavily occluded and can only serve as a static model.

#### 3.2 Level Creation

At the start of a Revit project, several levels are defined. These are horizontal planes that serve as a reference for the different 2D plans, views and objects. For instance, wall objects are typically placed in relation to the nearby levels, even if their geometry does not extend to this level. Most commonly, each storey of a building has its own level along with a top level for the roof and levels for the foundations and the ground level. However, the number



of levels is case-specific and can vary based on the type of project and building. For instance, a building with intermediate storeys can be designed with additional levels or the objects are modeled with a height offset depending on the application. In this research, we generate the levels semi-automatically. Given the classified floor mesh segments, the user can select any number of meshes and the levels are automatically created in the Revit environment through the Rhino.Inside API. The height for these levels are taken equal to the height of the centroid of the selected mesh component.

### 3.3 Walltype

The estimation of the walltype is performed using the Rhinocommon API as it has numerous functions for meshes. The goal of this step is to compute the most accurate LOD200 representation with respect to the observations. Therefore, the parameter extraction is weighted based on the surface area of the individual segments to ensure the best fit positioning of each partial wall.

**Height** Revit walls can either have a specific height or be placed with respect to the nearby levels. In this research we prefer the latter since the walls are then created dynamically and will move with the levels if the user adjusts them. The top and base constraint are selected based on the absolute vertical distance of the highest mesh point and the project levels (Eq.1).

$$\begin{aligned} \text{base constraint}_w &= l \in L, \text{ for which } \min |l_z - w_{z,\min}| \\ \text{top constraint}_w &= l \in L, \text{ for which } \min |l_z - w_{z,\max}| \end{aligned} \quad (1)$$

where  $l \in L$  are the levels defined in the project,  $l_z$  the height of a level,  $w \in W$  the walls in the project that each contain a set of segments  $\{s_1, s_2, \dots, s_i\}$  and  $w_{z,\min}$  and  $w_{z,\max}$  are respectively the lowest and highest Z-coordinate of the wall cluster. The base and top constraint do not have to be consecutive levels.

**Width** The width of an abstract LOD200 wall is defined as the uniform orthogonal distance between the two wall faces. In order to reduce the reconstruction error, the euclidean distance between the final wall faces and the observed mesh segments should be minimized. However, BIM models shouldn't unnecessarily be flooded with numerous walltypes. Also, the noise introduced by the sensors, registration and subsequent processing should be taken into consideration. We therefore propose a semi-automated wall width approach. First, the actual wall width is computed given the observations. In parallel, the user can create several walltypes within the Revit project according to their needs. Using the Rhinocommon API, the computed width is compared with the user defined width of each walltype and the best fit walltype is applied. This also allows the use of the Scan-to-BIM method to extend existing models. To compute the highest accurate distance from the observations, the computations are performed in 3D. This avoids any abstractions made in 2D methods and takes into account non-parallel wall segments, skewed walls and other shortcomings. The segments in both wall faces are sampled with a fixed number of points equal to their surface area. For every point on a segment, the dot product of that segment's normal and the vector of this point and its closest point on a nearby wall face, is computed. As this set is not normally distributed, we compute the mode of the set. As this is unusable for continuous distributions, we discretize the data by assigning frequency values to intervals of equal distance (Eq. 2).

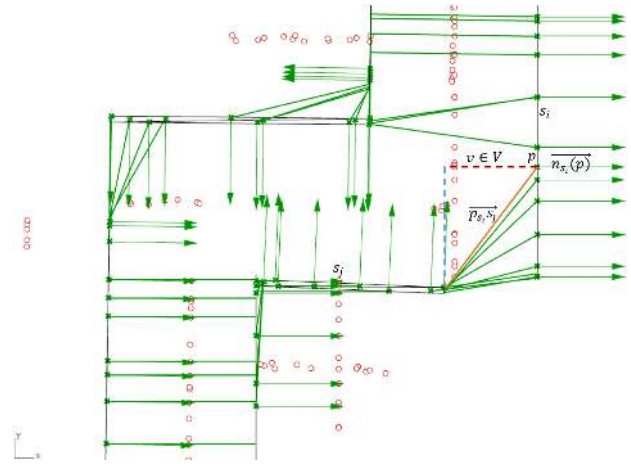


Figure 3: Width computation given  $s \in w$  (grey), with the sampled points  $\forall p \in s$  (green), the normals  $\overrightarrow{n_{s_i}(p)}$  per  $p$  (green arrows) and the vectors  $\overrightarrow{p s_j}$  connecting  $p$  to the nearest point on  $s_j$  (green lines).

$$\begin{aligned} \overrightarrow{P_{ij}} &= \left\{ \overrightarrow{p_i p_j} \mid \forall p_i \in s_i \wedge p_j \notin s_i : \underset{p_j}{\operatorname{argmin}} (\|p_i - p_j\|) \right\} \\ D &= \left\{ \overrightarrow{p_{ij}} \cdot \overrightarrow{n_{s_i}(p_i)} \mid \forall \overrightarrow{p_{ij}} \in \overrightarrow{P_{ij}} \wedge |\overrightarrow{p_{ij}} \cdot \overrightarrow{n_{s_i}(p_i)}| \geq t_{copl} \right. \\ &\quad \left. \wedge |\overrightarrow{n_{s_j}(p_j)} \cdot \overrightarrow{n_{s_i}(p_i)}| \geq t_{par} \wedge \|\overrightarrow{p_{ij}}\| \leq t_d \right\} \\ d_w &= Mo(|D|) \end{aligned} \quad (2)$$

where  $\forall p \in s$  are the sampled points on each segment  $s \in w$ ,  $\overrightarrow{n_{s_i}(p_i)}$  the normal of mesh segment  $s_i$  at the location of  $p$ . By sampling the surfaces with respect to their surface area, the actual width  $d_w$  is more accurate since the method favors large wall surfaces which have a significant impact on the reconstruction accuracy. The walltype is then decided by the minimal difference between the Revit walltype width  $d_r$  and  $d_w$  (Eq. 3).

$$\begin{aligned} \text{walltype}_w &= \text{walltype}_r \in \text{Revit walltypes, for which} \\ &\quad \min |d_w - d_r| \end{aligned} \quad (3)$$

### 3.4 Wall centre surface

Revit Basic Walls can be either slanted or vertical structures. The vertical variant is created from a polyline in a horizontal plane. It is the most commonly used geometry in as-built models despite its inherent abstraction. The alternative is a slanted basic wall which is placed based on a complex surface which in Revit is referred to as the wall face. The polyline and the wall face are of course linked since the polyline is the projection of the wall face in the case of a vertical wall. In this research, we consider this face or polyline to be the centre of the wall and will be referred to the centre surface and the centreline to avoid confusion with the wall faces on either side of the wall. Also, the emphasis is on the creation of vertical walls since slanted walls are typically unwieldy for the majority of applications. The centreline is constructed as follows. First, a set of 2D points is created based on the sampled points  $P$  of the mesh segments, their normals and  $d_w$  computed in the previous paragraph (Eq. 4).

$$\begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} = \begin{bmatrix} P_x + 0.5d_w \overrightarrow{n_s(P)} \hat{D} \\ P_y + 0.5d_w \overrightarrow{n_s(P)} \hat{D} \\ l_z \end{bmatrix} \quad (4)$$

after which  $\forall c \in C$  are fed to a least squares and conditional RANSAC fitting algorithm that establishes the centreline. Three types of centrelines are identified: straight lines, arcs and poly-lines. For each of the cases, RANSAC computes

$$1 - p = (1 - \omega^n)^k \quad (5)$$

$k$  number of fits where the number of selected points  $N$  is condition per type based on the dimensions of  $w$  in X and Y.  $\omega$  is ratio of inliers and is considered fairly high since  $C$  is sampled solely based on wall observations (Eq. 6).

$$N = \begin{cases} \text{Line : } (\#n = 2 \|c_i - c_j\| \geq 0.5 \text{Dim}_{XY}(w)) \\ \text{Arc : } (\#n = 3 \|c_i - c_j\| \geq 0.2 \text{Dim}_{XY}(w)) \\ \text{Polyline : } (\#n > 3 \|c_i - c_j\| \geq \frac{\text{Dim}_{XY}(w)}{\#n}) \end{cases} \quad (6)$$

The best fitting model  $\psi_w$  is chosen based on the error between  $P$  in the XY plane and  $\psi(N)$  given half of the final wall thickness  $d_w$ .

$$\text{inliers}(\psi(N)) = \left\{ p \in P \mid \|\psi(N) - p\| - 0.5d_w \leq t_o \right\} \quad (7)$$

$$\psi_w = \underset{\psi(N)}{\text{argmin}}(|\text{inliers}(\Psi)|)$$

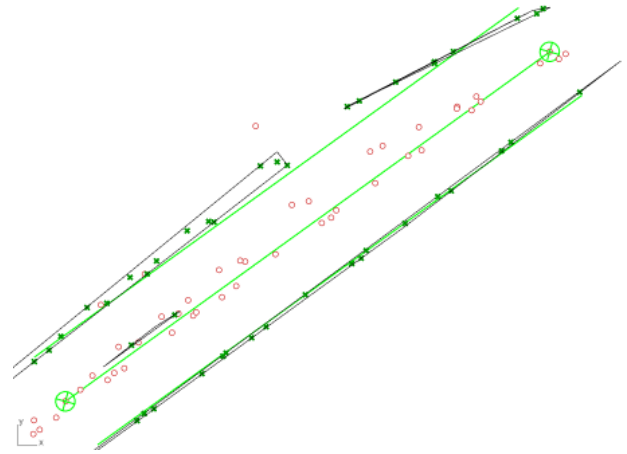
In the case of walls that are only observed from one side, we compute the wall face instead of the centreline. Similar to the two-faced walls, we compute the trace based on sampled points  $C$  but generate them with the normal pointing away from the nearest floor. A default thickness of 100mm is given to these walls.

To increase the efficiency, not every model is tested to compute the best fit model. Based on the expected inliers, the first function to meet the inlier threshold is kept. If the maximum number of control points is met first, the model with the highest score is kept.

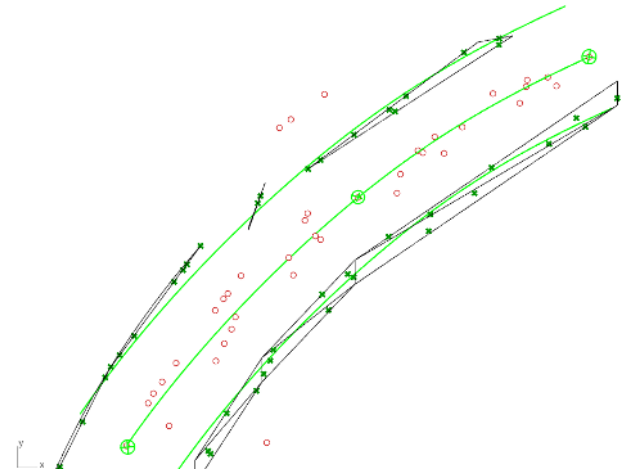
Once all the relevant parameters have been extracted, the geometry of the partial walls is created. Using the Rhino.inside API, a Revit Basic wall entity is created for every  $w$ , with the appropriate walltype, base and top constraint and centreline.

### 3.5 Wall topology

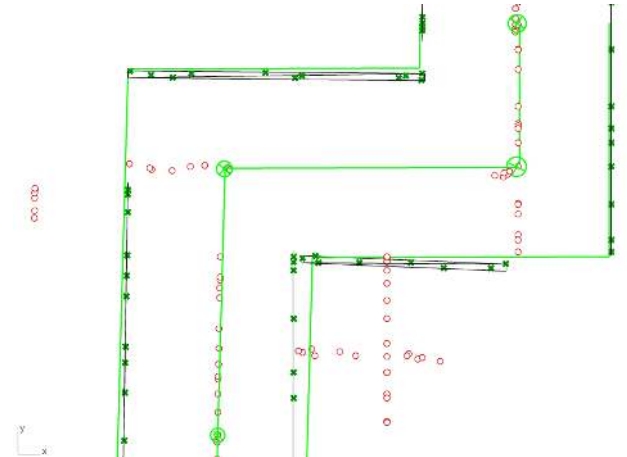
In order to create a logic building model, the topology of the partial walls should be adjusted. As Revit native basic walls are created, the semi-automated toolkit of the Revit API can be used for this operation. Using the intuitive *trim*, *extend* and *join* functions, a user can efficiently finalize the model while performing a quality control. In future work, we will integrate an automated topology creation method but currently this method is preferred since wall connections can be established in several ways and are often project and modeler specific.



(a) Straight line:  $\#n = 2$  model,  $\bar{D} = 1.14m$ ,  $Mo(D) = 1.22m$ ,  $d_{w,manual} = 1.20m$ ,  $time = 0.14s$



(b) Arc:  $\#n = 3$  model,  $\bar{D} = 1.41m$ ,  $Mo(D) = 1.52m$ ,  $d_{w,manual} = 1.40m$ ,  $time = 0.26s$



(c) Polyline:  $\#n = 4$  model,  $\bar{D} = 1.77m$ ,  $Mo(D) = 1.80m$ ,  $d_{w,manual} = 1.80m$ ,  $time = 1.3s$

Figure 4: Model fitting of  $\psi(N)$  (green lines) based on  $C$  (red dots).  $\|\psi(N) - P\|$  is computed between  $P$  (green dots) and  $C$  given  $0.5d_w$  (offsetted green lines).

## 4. EXPERIMENTS

The proposed algorithm is successfully implemented in Revit using Matlab and the Rhinocommon and Revit API. Figure 5 shows

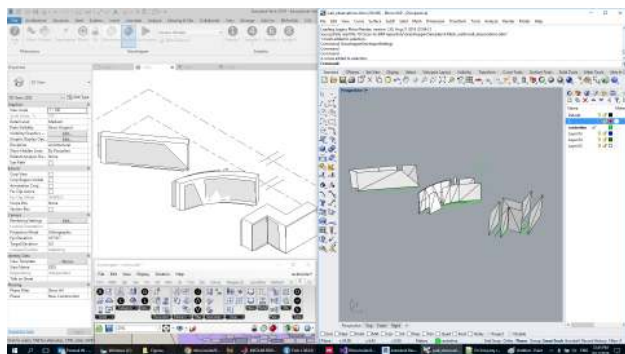


Figure 5: Overview of the three wall cases. The interface of Revit and Rhino showcase the observed and reconstructed wall geometry which are linked through the Rhino.Inside Plug-in.

an example of the interface between both software and the components linking to the Matlab functions. Both the level creation, walltype identification and centreline estimation were all performed fully automatically. To test the algorithm's performance, the centreline computation is tested on 3 distinct wall cases i.e. a straight wall, a curved wall and a wall with a discontinuous wall axis. Each case is simulated with a number of mesh observations which is the output from our previous work (Bassier and Vergauwen, 2019a). The observed segments represent realistic conditions with non-parallel meshes, occlusions, non-vertical segments, non-uniform thickness and clutter. Overall, the segments are littered with frequently occurring shortcomings given a mostly successful clustering of the wall segments. For the sake of the experiment, the scale of the walls was increased to augment the visibility and the magnitude of the errors. The automatically reconstructed geometry is compared to manually designed walls. The same wall-type was used and the sampled points  $P$  on the observed mesh segments is used as a reference.

The results of the comparison are shown in Table 1. The manual geometry was established with an accuracy of respectively 0.13m, 0.19m and 0.12m for the line, arc and polyline. The automated procedure achieved a similar accuracy (0.12m, 0.11m and 0.18m). Moreover, the type of model selected by the automated procedure closely aligns with the actual shape of the wall. As discussed in the methodology, the first model to meet the inlier threshold is kept. Table 1 shows clear spikes in inliers when the appropriate model is fit and thus this is considered a reliable indicator. Default values are provided for the fully automated procedure but users can influence the expected inliers, distance threshold and the number of control points. Overall, the method shows promising results for the reconstruction of wall geometry even in complex environments.

## 5. CONCLUSION

This paper presents an unsupervised method to reconstruct wall geometry from unstructured point clouds of buildings. The method takes as input clustered wall segments originating from previous works and outputs Native Revit and Rhino LOD200 BIM objects. Given sampled point on the observed segments, the method computes the best fit centreline that serves as the basis for the reconstructed entities. Additionally, each wall is associated with the appropriate Revit walltype in order to create consistent wall geometry. The result is a set of partial walls that closely align with the observed faces of the wall. Once the partial geometry is established, the wall topology can be semi-automatically adjusted in the Revit software to create a truthful as-built BIM.

The experiments indicate that the used method is a promising reconstruction framework. The achieved accuracy and model selection rival that of manual modeling by experts for a variety of scenes. The procedure is robust to noise and clutter and deals with frequently occurring problems such as non-parallel walls. In future work, the wall topology will also be automated to even further reduce the user effort to create as-built BIM geometry.

## 6. ACKNOWLEDGEMENTS

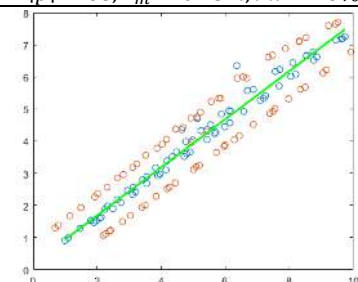
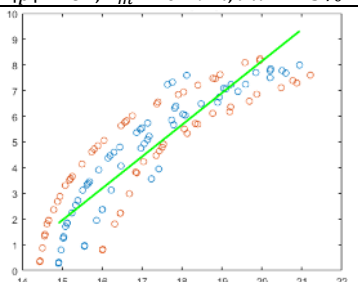
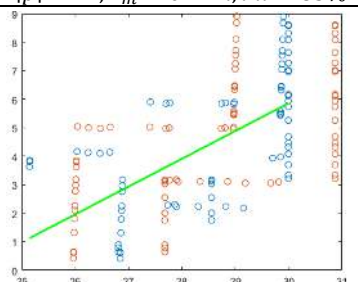
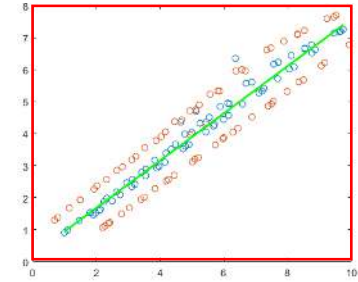
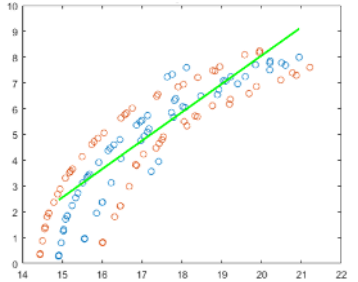
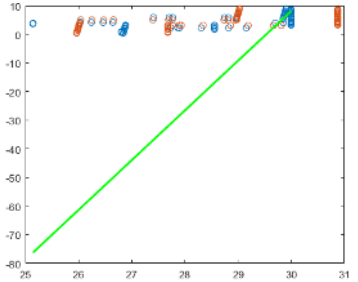
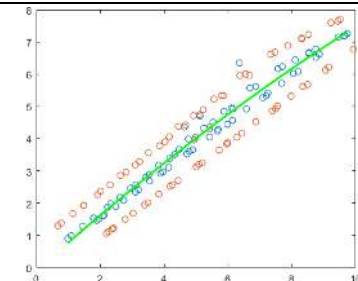
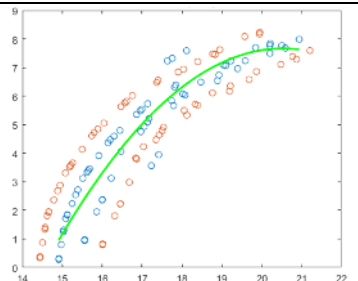
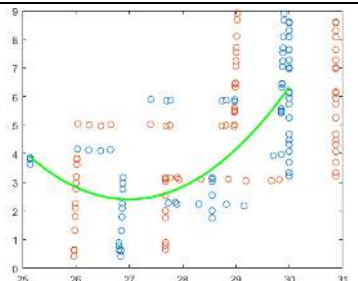
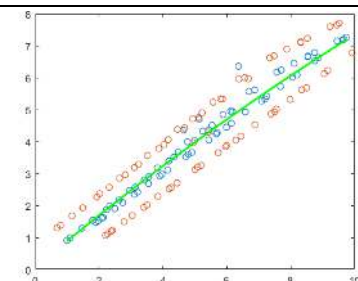
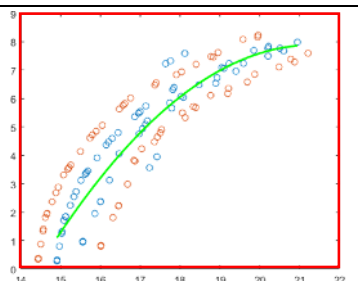
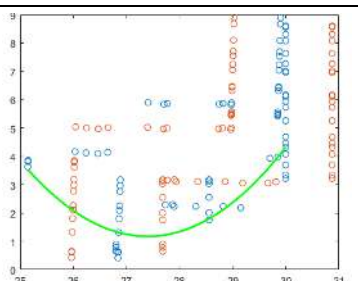
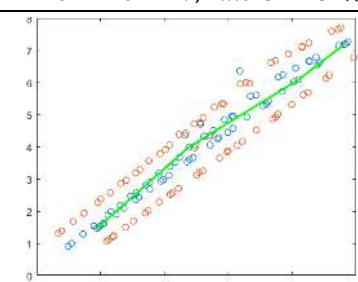
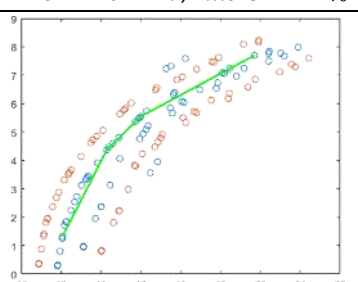
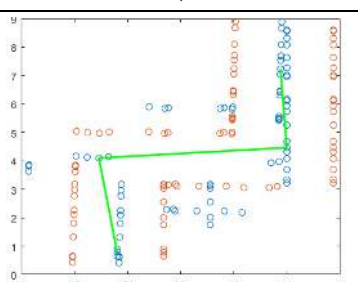
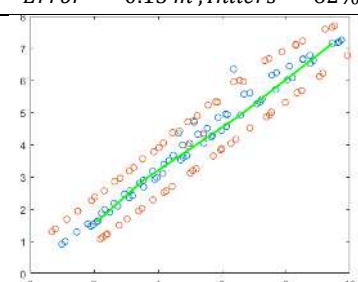
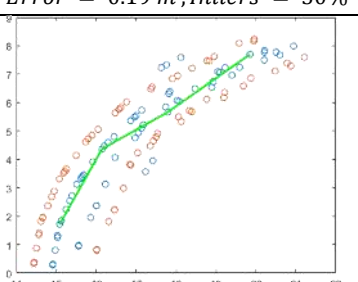
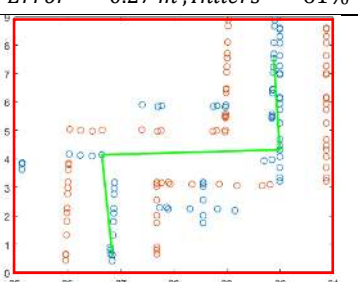
This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement 779962), the FWO research foundation (under FWO PhD SB fellowship 1S87120N) and the Geomatics research group of the Department of Civil Engineering, TC Construction at the KU Leuven in Belgium.

## REFERENCES

- Adan, A. and Huber, D., 2011. 3D Reconstruction of Interior Wall Surfaces under Occlusion and Clutter. *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission* pp. 275–281.
- Anagnostopoulos, I., Patraucean, V., Brilakis, I. and Vela, P., 2016. Detection of walls, floors and ceilings in point cloud data. In: *Construction Research Congress 2016*.
- Barazzetti, L., 2018. Point cloud occlusion recovery with shallow feedforward neural networks. *Advanced Engineering Informatics*.
- Bassier, M. and Vergauwen, M., 2019a. CLUSTERING OF WALL GEOMETRY FROM UNSTRUCTURED POINT CLOUDS. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII(February)*, pp. 6–8.
- Bassier, M. and Vergauwen, M., 2019b. Clustering of Wall Geometry from Unstructured Point Clouds Using Conditional Random Fields. *Remote Sensing*.
- Bassier, M. and Vergauwen, M., 2019c. Clustering of Wall Geometry from Unstructured Point Clouds Using Conditional Random Fields. *Remote Sensing* 42(2/W9), pp. 101–108.
- Bassier, M., M. B., Van Genechten, B. and M. V., 2017. Octree-Based Region Growing and Conditional Random Fields. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-2/W8, 2017 5th International Workshop LowCost 3D – Sensors, Algorithms, Applications XLII(November)*, pp. 28–29.
- Bassier, M., Van Genechten, B., Vergauwen, M., Genechten, B. V. and Vergauwen, M., 2018. Classification of sensor independent point cloud data of building objects using random forests. *Journal of Building Engineering* (April), pp. 1–10.
- Bassier, M., Vergauwen, M. and Van Genechten, B., 2016. Automated Semantic Labelling of 3D Vector Models for Scan-to-BIM. *4th Annual International Conference on Architecture and Civil Engineering (ACE 2016)* (April), pp. 93–100.
- BIMForum, 2016. Level of Development Specification. Technical report, BIMFORUM.
- Bosché, F., Ahmed, M., Turkan, Y., Haas, C. T. and Haas, R., 2014. The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components. *Automation in Construction* 49, pp. 201–213.



Table 1: Result of the automated hearthline reconstruction. Each column represents the fitting results for the Line, Arc and Polyline. The orange dots are P, the blue dots are C and the green lines are the computed hearthline given the different methods. The red rectangle depicts the first model to fulfill the expected error and inlier rate.

	Line $ p  = 68, E_m = 0.13m, inl = 26\%$	Arc $ p  = 62, E_m = 0.19m, inl = 23\%$	Polyline $ p  = 77, E_m = 0.12m, inl = 85\%$
Fit Line (TLS)	 <i>Error = 0.13 m, Inliers = 59%</i>	 <i>Error = 0.41 m, Inliers = 16%</i>	 <i>Error = 0.69 m, Inliers = 16%</i>
Fit Line (Ransac)	 <i>Error = 0.12 m, Inliers = 71%</i>	 <i>Error = 0.41 m, Inliers = 34%</i>	 <i>Error = 0.98 m, Inliers = 48%</i>
Fit Arc (TLS)	 <i>Error = 0.13 m, Inliers = 65%</i>	 <i>Error = 0.23 m, Inliers = 44%</i>	 <i>Error = 0.60 m, Inliers = 12%</i>
Fit Arc (Ransac)	 <i>Error = 0.12 m, Inliers = 87%</i>	 <i>Error = 0.11 m, Inliers = 71%</i>	 <i>Error = 0.41m, Inliers = 16%</i>
Fit Poly (Ransac) $n = 4$	 <i>Error = 0.13 m, Inliers = 62%</i>	 <i>Error = 0.19 m, Inliers = 50%</i>	 <i>Error = 0.27 m, Inliers = 61%</i>
Fit Poly (Conditioned Ransac) $n = 4$	 <i>Error = 0.14 m, Inliers = 60%</i>	 <i>Error = 0.20 m, Inliers = 48%</i>	 <i>Error = 0.18 m, Inliers = 62%</i>

- Bosché, F., Turkan, Y. and Haas, C., 2013. Tracking the Built Status of MEP Works : Assessing the Value of a Scan-vs. -BIM System. *Journal of Computing in Civil Engineering*.
- Budroni, A. and Böhm, J., 2010. Automatic 3d Modelling Of Indoor Manhattan-world Scenes From Laser Data. *ISPRS Commission V Mid-Term Symposium: Close Range Image Measurement Techniques XXXVIII*, pp. 115–120.
- BuildingSMART International Ltd, 2013. Industry Foundation Classes Release 4 (IFC4).
- Díaz-Vilarino, L., Khoshelham, K., Martínez-Sánchez, J. and Arias, P., 2015. 3D modeling of building indoor spaces and closed doors from imagery and point clouds. *Sensors* 15(2), pp. 3491–3512.
- Dittrich, A., Weinmann, M. and Hinz, S., 2017. Analytical and numerical investigations on the accuracy and robustness of geometric features extracted from 3D point cloud data. *ISPRS Journal of Photogrammetry and Remote Sensing* 126, pp. 195–208.
- Garstka, J. and Peters, G., 2016. Evaluation of Local 3-D Point Cloud Descriptors in Terms of Suitability for Object Classification. *13th International Conference on Informatics in Control, Automation and Robotics 2(Icinco)*, pp. 540–547.
- Gimenez, L., Robert, S., Suard, F. and Zreik, K., 2015. Automatic reconstruction of 3D building models from scanned 2D floor plans. *Automation in Construction* 63(2016), pp. 48–56.
- Hichri, N., Stefani, C., Luca, L. D. and Veron, P., 2013. Review of the As-Built Bim Approaches. *3D-ARCH 2013 - 3D Virtual Reconstruction and Visualization of Complex Architectures*.
- Kavanaugh, J. C. K. S. P. E. B., 2013. Client-Guide-to-3D-Scanning-and-Data-Capture. Technical report, Building Information Modelling Task Group.
- Landrieu, L., Mallet, C. and Weinmann, M., 2017. Comparison of belief propagation and graph-cut approaches for contextual classification of 3D lidar point cloud data. *IGARSS'2017*.
- Macher, H., Landes, T. and Grussenmeyer, P., 2017. From Point Clouds to Building Information Models: 3D Semi-Automatic Reconstruction of Indoors of Existing Buildings. *Applied Sciences*.
- Michailidis, G. T. and Pajarola, R., 2016. Bayesian graph-cut optimization for wall surfaces reconstruction in indoor environments. *Visual Computer* pp. 1–9.
- Mura, C., Mattausch, O. and Pajarola, R., 2016. Piecewise-planar Reconstruction of Multi-room Interiors with Arbitrary Wall Arrangements. *Computer Graphics Forum* 35(7), pp. 179–188.
- Mura, C., Mattausch, O., Jaspe Villanueva, A., Gobbetti, E. and Pajarola, R., 2014. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. *Computers & Graphics* 44, pp. 20–32.
- Nguyen, A. and Le, B., 2013. 3D point cloud segmentation : A survey 3D. *2013 6th IEEE Conference* (November), pp. 225–230.
- Nikoohemat, S., Peter, M., Oude Elberink, S. and Vosselman, G., 2017. Exploiting Indoor Mobile Laser Scanner Trajectories for Semantic Interpretation of Point Clouds. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences IV-2/W4*(September), pp. 355–362.
- Ochmann, S., Vock, R., Wessel, R. and Klein, R., 2016. Automatic reconstruction of parametric building models from indoor point clouds. *Computers & Graphics* 54, pp. 94–103.
- Oesau, S., Lafarge, F. and Alliez, P., 2014. Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. *ISPRS Journal of Photogrammetry and Remote Sensing* 90, pp. 68–82.
- Patrucean, V., Armeni, I., Nahangi, M., Yeung, J., Brilakis, I. and Haas, C., 2015. State of research in automatic as-built modelling. *Advanced Engineering Informatics* 29, pp. 162–171.
- Previtali, M., Barazzetti, L., Brumana, R. and Scaioni, M., 2014. Towards automatic indoor reconstruction of cluttered building rooms from point clouds. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences II-5*(June), pp. 281–288.
- Qi, C. R., Su, H., Mo, K. and Guibas, L. J., 2016. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *arXiv preprint arXiv:1612.00593*.
- Robert McNeel & Associates, 2019. Rhino.Inside.
- Tang, P., Huber, D., Akinci, B., Lipman, R. and Lytle, A., 2010. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in Construction* 19(7), pp. 829–843.
- Thomson, C. and Boehm, J., 2015. Automatic geometry generation from point clouds for BIM. *Remote Sensing*.
- Turner, E. and Zakhori, A., 2014. Floor Plan Generation and Room Labeling of Indoor Environments from Laser Range Data. *GRAPP, International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications* pp. 1–12.
- Valero, E., Adán, A. and Cerrada, C., 2012. Automatic method for building indoor boundary models from dense point clouds collected by laser scanners. *Sensors (Switzerland)*.
- Vo, A. V., Truong-Hong, L., Laefer, D. F. and Bertolotto, M., 2015. Octree-based region growing for point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing* 104, pp. 88–100.
- Volk, R., Stengel, J. and Schultmann, F., 2014. Building Information Modeling (BIM) for existing buildings - Literature review and future needs. *Automation in Construction* 38, pp. 109–127.
- Weinmann, M., Jutzi, B., Mallet, C. and Weinmann, M., 2017. GEOMETRIC FEATURES and THEIR RELEVANCE for 3D POINT CLOUD CLASSIFICATION. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. 4number 1W1.
- Wolf, D., Prankl, J. and Vincze, M., 2015. Fast Semantic Segmentation of 3D Point Clouds using a Dense CRF with Learned Parameters. *IEEE International Conference on Robotics and Automation (ICRA)*.
- Xiong, X., Adan, A., Akinci, B. and Huber, D., 2013. Automatic creation of semantically rich 3D building models from laser scanner data. *Automation in Construction* 31, pp. 325–337.