Point Cloud to BIM Using Python Codes

Slávek Zbirovský*[1] and Václav Nežerka[1]

[1]Czech Technical University in Prague, Faculty of Civil Engineering, Department of Physics
Thákurova 7,166 29 Prague 6, Czech Republic
*Corresponding author: slavek.zbirovsky@fsv.cvut.cz

13th May 2024

Abstract

The construction industry has witnessed a significant development in digitalization, leading to the emergence of Construction 4.0. This term refers to integrating modern digital technologies to enhance project efficiency, quality, safety, and sustainability. In light of the growing emphasis on ecological responsibility, there is an increasing interest in revitalising ageing structures to meet the current sustainability and energy efficiency standards. Proposed solutions to the challenges of ageing infrastructure include renovation, modernization, and deconstruction with material recycling. Nevertheless, the lack of adequate documentation of existing buildings represents a significant obstacle. The utilization of point cloud technology through laser scanning or digital photogrammetry provides a solution, facilitating the creation of accurate three-dimensional models. While scan-to-BIM processes are traditionally lengthy due to the manual work of people, semi-automatic and automatic segmentation solutions are emerging. These solutions aim to streamline the conversion of point clouds to Building Information Models (BIM), improving efficiency and accuracy in the modelling process. This research presents an open-source method for classifying, segmenting, and reconstructing fully volumetric 3D models from the point cloud into Industry foundation classes (IFC). The openBIM format ensures that the data can be used in various software applications.

Keywords: Scan-to-BIM, Point cloud processing, Segmentation, Python, IFC

1. Introduction

A point cloud dataset is a three-dimensional representation of a scanned object, generated using technologies such as laser scanning (LiDAR), photogrammetry, or similar methods. These geodetic surveys capture the surface of the scanned object by creating a dense network of points, providing detailed spatial information. Point cloud technology finds diverse applications across various fields such as architecture, engineering, construction, and archaeology. Its ability to accurately capture and represent real-world objects makes it an invaluable digital preservation, modelling, and analysis tool. However, the high precision of the collected data comes with significant demands on 3D modelling software to handle these large datasets effectively. For this reason, point cloud data is often converted into 3D parametric models, which facilitate the manipulation of the attributes of individual parts. During this process, point clouds and building elements such as slabs, walls, windows, and doors are overlaid in the modelling software close to one another. This process is often referred to as scan-to-BIM, and in recent years, it has been conducted primarily through manual modeling.

However, manual modelling processes are time-consuming and labor-intensive, requiring skilled personnel to interpret and convert the point cloud data accurately. Automated methods for Point Cloud to Building Information Modeling conversion have been developed to streamline and expedite this process. Automated conversion reduces the time and effort required and enhances the accuracy and consistency of the resulting BIM models. These automated methods utilize algorithms and software tools to interpret the point cloud data and generate parametric BIM models directly from the scanned data. Automating the conversion process helps overcome the challenges associated with manual modelling, making the creation of BIM models from point cloud data more efficient and accessible.

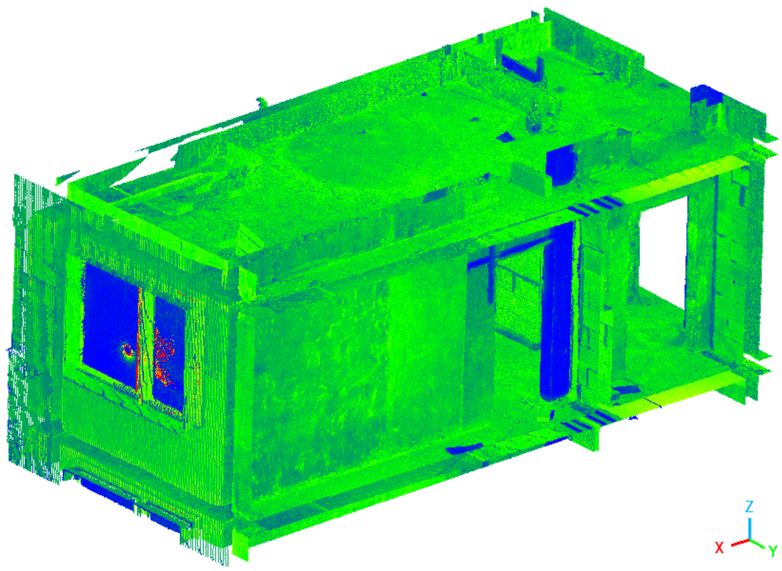

Figure 1: Point cloud cropped from the large dataset for algorithm testing (Zbirovský, 2024).

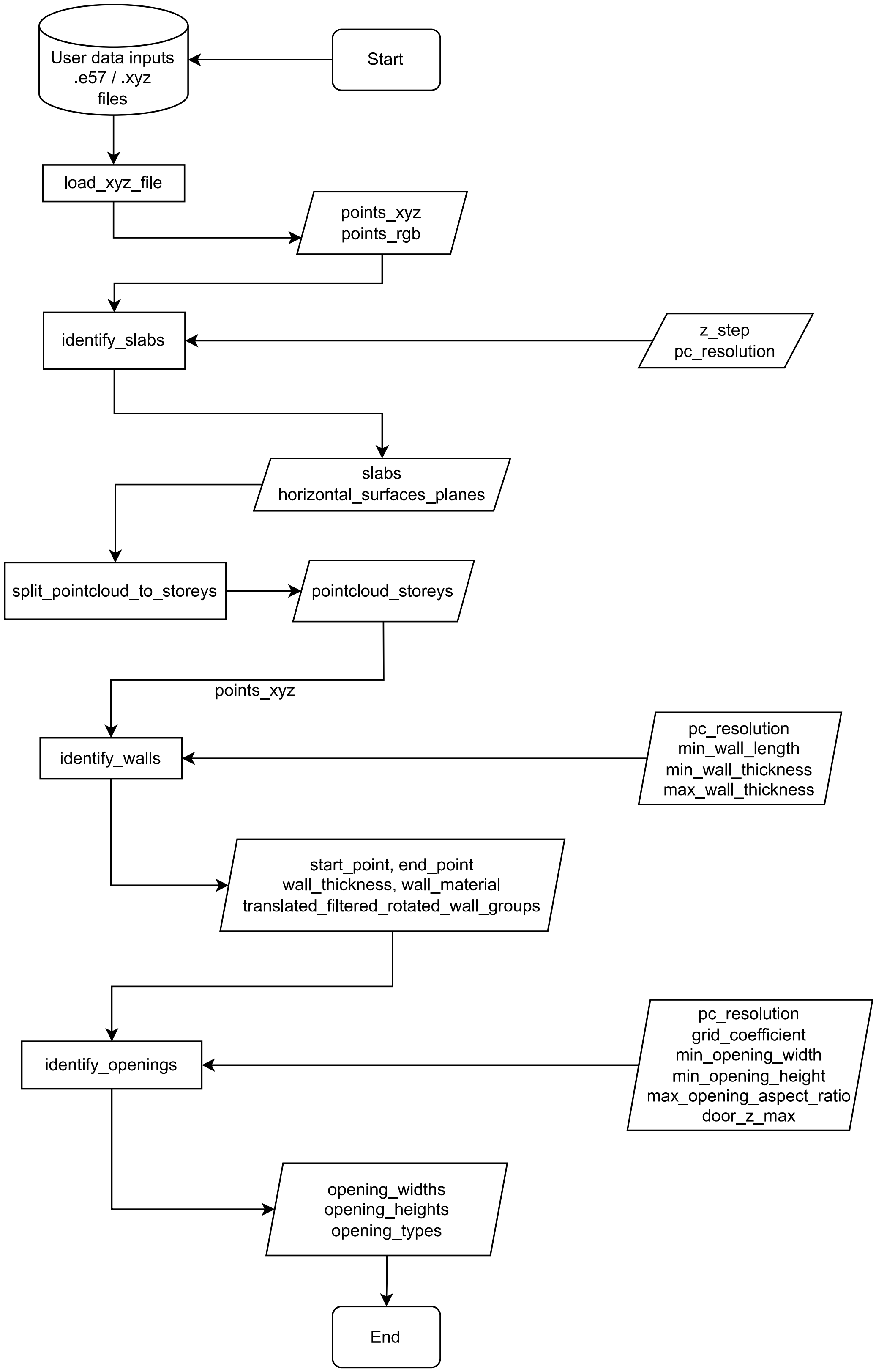

Figure 2: Program flowchart illustrating the algorithm for converting unorganized point clouds into a digital model of a building. The flowchart depicts the code implementation's inputs, outputs, and sequential steps.

2. Methods

The process can be divided into the following steps: (i) slab segmentation, (ii) floor segmentation, (iii) wall segmentation, and (iv) opening detection. Furthermore, a flowchart detailing the code implementation is presented in Figure 2 to provide a more comprehensive understanding of the algorithm.

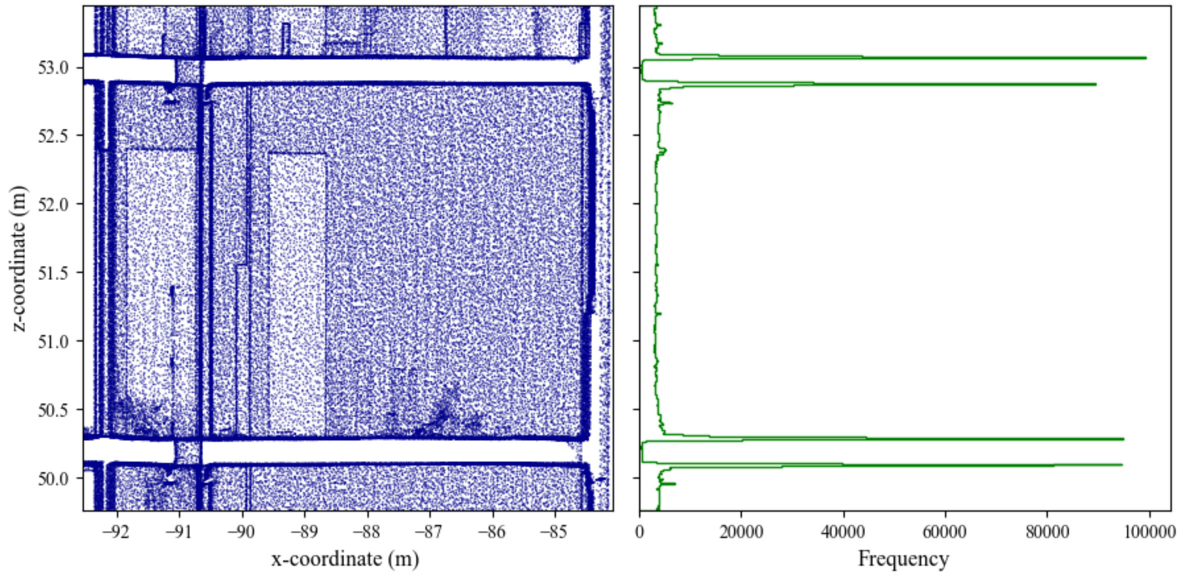

Figure 3: Point cloud cross-section (on the left). A z-coordinate histogram of point cloud data (on the right) (Zbirovský, 2024).

2.1 Slabs

The segmentation of the floor slabs is the first and most logical step, which is also carried out in the normal manual scan-to-BIM process. The segmentation will divide the building into smaller horizontal units that can be analyzed in more detail. The prerequisite for segmentation is that the number of points in the horizontal planes greatly exceeds the number of other points. Figure 3 shows this phenomenon in the height histogram for one floor with corresponding point cloud cross-section. Elevation planes that meet a point quantity condition, set by a threshold defined as a percentage of the maximum, are extracted and paired with their opposing surfaces based on their surface index. These elevation planes establish parameters defining the ceiling slab and its vertical position. A two-dimensional point cloud is generated by merging surfaces and eliminating redundant coordinates. This point cloud is subsequently analysed to obtain a polygon defining the floor plan shape of the slab.

Compared to the first version dedicated to the ceiling slabs (Zbirovský & Nežerka, 2024), the algorithm to obtain the polygon is optimized due to the high computation cost. The analysis is now performed on the binarized image of the floor plan. The largest contour is extracted, representing the slab's floor plan polygon. To reduce the number of points that represent the resulting polygon, the perimeter contour is smoothed using the Douglas-Peuckert function it takes the x- and y-coordinates of the contour, along with a parameter epsilon, which determines the maximum distance from the approximated line for a point to be considered on that line. The function returns the smoothed x- and y-coordinates of the contour, and the smoothed contour itself is represented as an array. The epsilon parameter is critical in determining the level of simplification applied to a curve. Epsilon controls the trade-off between retaining detail and achieving a more generalized representation by specifying the maximum allowable distance between the original and simplified curves. A smaller epsilon value results in a simplified curve closely following the original, while a larger epsilon value produces a more generalized, simplified curve.

2.2 Walls

In recent years, two distinct approaches have emerged for the classification and segmentation of walls: (i) The "Room-based" approach emphasizes the geometry of the room and identifies uniform surrounding surfaces with a preset thickness (Mura et al., 2016; Tran et al., 2019; Yang et al., 2019). In contrast, (ii) the "Volumetric-based" approach focuses on individual structural elements that volumetrically define the junction of opposing surfaces (Bassier & Vergauwen, 2020; Martens & Blankenbach, 2023; Ochmann et al., 2019). For BIM applications, the volumetric approach is preferred.

Simplifying assumptions are often employed for modeling walls. One of the most well-known is the "Manhattan world" assumption, which allows for searching for walls in only two main directions, typically along the vertical and horizontal axes. Algorithms using this assumption are highly robust but are also significantly limited in terms of the complexity of the scenes they can accurately represent. However, within the scope of this work, it was possible to avoid this simplifying condition. Conversely, a constraint that had to be accepted limited the geometry of the walls to only flat ones with constant thickness. This condition is sometimes called the "Atalanta world" assumption, where scenes can be described by vertical and horizontal planes in 3D. Importantly, this assumption doesn't demand that vertical planes be perpendicular to each other, offering greater flexibility in scene representation (Pintore et al., 2020).

The first step in wall detection is to prepare the data. The analysis is performed on each floor within a height range of 90–100%. Points within this height range are projected onto the x-y plane, and a binarized mask is created based on these points. Clear boundaries are formed where walls intersect the ceiling structure, as there are no points in these areas that correspond to the surfaces of the walls. The *'cv2.findContours'* function is then applied to this binary image to identify contours, where the white area represents objects or regions of interest. These contours are then extracted to line segments. In the next step, these line segments are connected if they are collinear and merged into groups if they are parallel. The last step is to find an axis and define a start and end point. The axis position should minimize the total distance from the line segments, thus

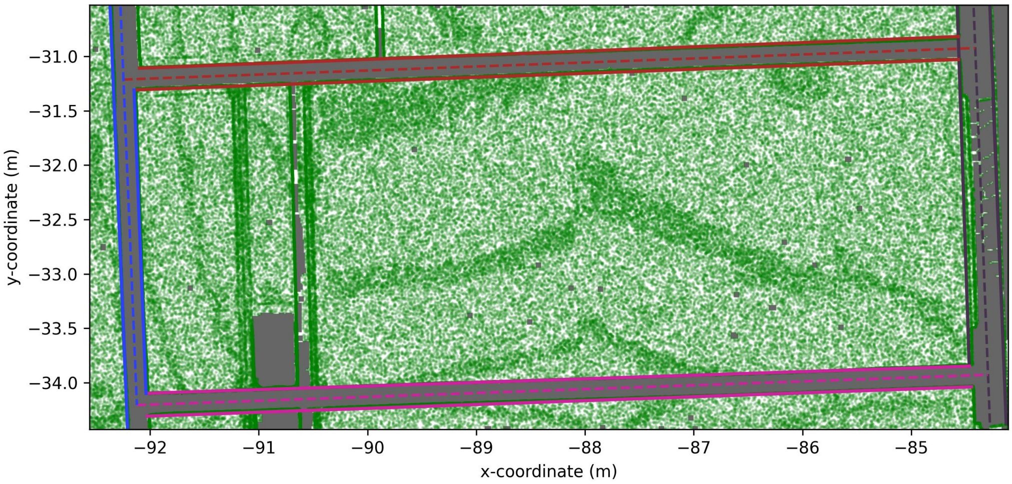

Figure 4: Walls (colored) segmented from point cloud (green), represented with their axes (Zbirovský, 2024).

accurately representing the group's alignment. Figure 4 displays the final results from the wall segmentation function.

2.3 Openings

Several methods have emerged over the years for the detection of openings. For doors, solutions have been divided into histogram-based solutions (Chen, 2022; Mahmoud et al., 2024) and solutions utilizing scanner trajectory (Nikoohemat et al., 2018). However, each method has its drawbacks. The density-based analysis method is highly sensitive to fluctuations caused, for example, by furniture covering the wall's surface. The second method, based on trajectory, is highly restrictive regarding the data collection technology.

The histogram-based approach is complemented by heuristic rules for identifying door and window openings. These rules include, for example, (i) the position of the sill to distinguish between window and door openings, (ii) the aspect ratio of the assessed opening, and (iii) the maximum and minimum width of the opening. One limitation of the output generated by this type of analysis is the shape, which is restricted to rectangular openings. Figure 5 depicts a side view of the exterior wall that has been evaluated. Histograms are displayed to validate the accuracy of the operations. These histograms represent the density of the point cloud and the set threshold values (indicated by dashed lines).

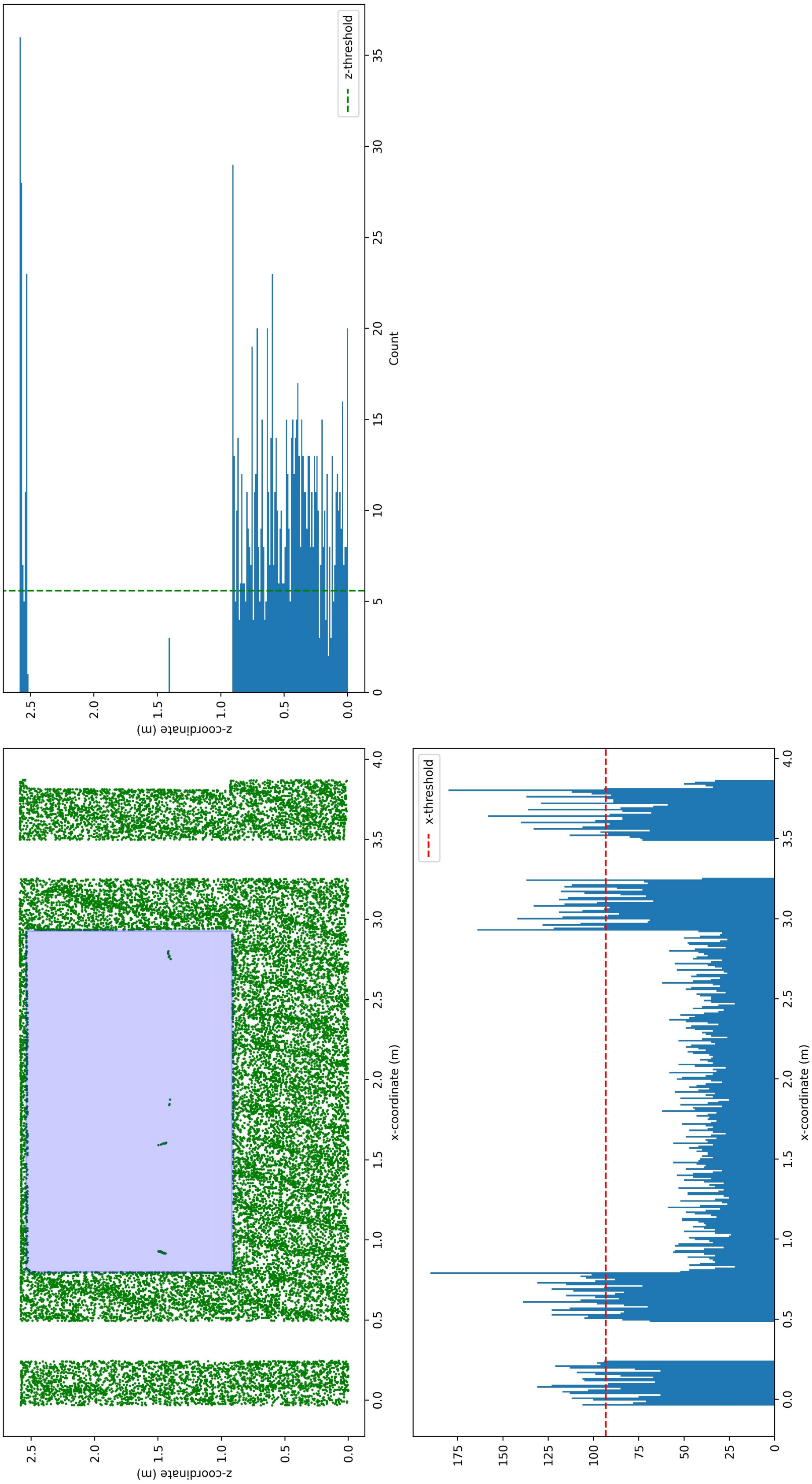

Figure 5: Detection of wall openings using point density analysis. Histograms of point cloud density along the z-coordinate (top left) and x-coordinate (bottom right), along with a side view of the evaluated wall point cloud (bottom left) showing the detected window opening (highlighted in blue) (Zbirovský, 2024).

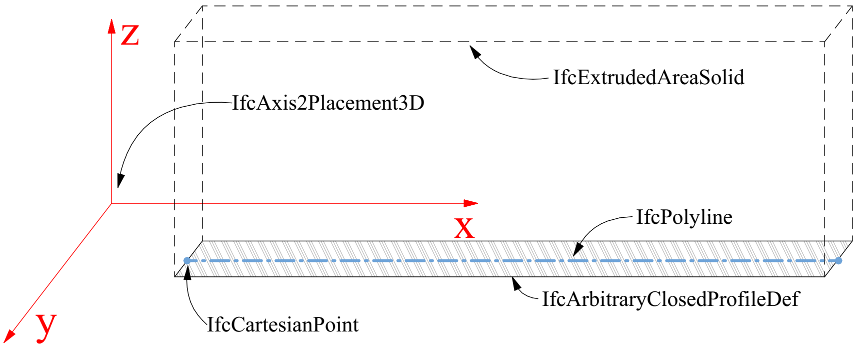

Figure 6: Main IFC classes used to create a geometric representation of the wall (Zbirovský, 2024).

2.4 Geometry reconstruction into IFC

The industry foundation classes file format was developed and is primarily intended for data transfer between the various participants in the BIM process. The building information model and its associated information are shared during this data transfer. The use of IFC as the end format after transferring the point cloud to the BIM model is not a new concept; a similar principle was implemented by Martens & Blankenbach (2023), whose solution involves the reconstruction of slabs, floors, and walls.

The output in the IFC 4 ADD2 TC1 format was generated using the IfcOpenShell library (IfcOpenShell, 2023) within a Python script. This step enabled the definition of all necessary attributes of individual IFC classes. The DesignTransferView_V1.0 was selected as the Model View Definition (MVD) to allow for potential further editing within BIM software. This choice ensured compatibility and facilitated seamless integration with BIM platforms such as ArchiCAD, Revit, and Allplan.

A spatial hierarchy has been established to facilitate the straightforward inclusion of building elements into their respective locations. This hierarchical structure reflects the project's common organization. In each IFC file, precisely one instance of *'IfcProject'* or another *'IfcContext'*, such as *'IfcProjectLibrary'*, must be defined. All other data within the file is then linked to this primary IFC class. *'IfcProject'* defines crucial attributes such as unit definitions, project name, project phase, and more.

In IFC, building elements are represented by specific entity classes. For instance, the *'IfcSlab'* class represents ceiling slabs. The geometric representation of a ceiling slab is achieved through a swept solid approach. The slab's perimeter is defined by an *'IfcArbitraryClosedProfileDef'*, typically a polyline. This profile is then extruded along a specified axis using the *'IfcExtrudedAreaSolid'* entity. Furthermore, parameters such as extrusion depth and local placement within the building are also defined.

Similarly, walls are represented by the *'IfcWall'* class. Their geometry, defined by start and end points using the *'IfcCartesianPoint'* class, is extruded using the *'IfcArbitraryC-*

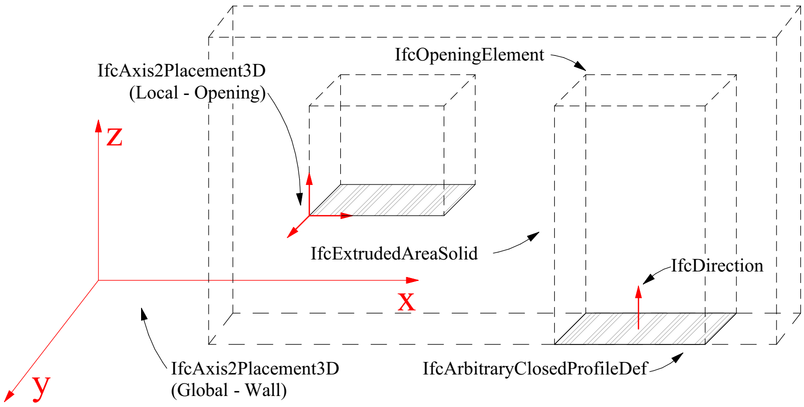

Figure 7: Main IFC classes used to create a geometric representation of opening. (Zbirovský, 2024).

losedProfileDef' and *'IfcExtrudedAreaSolid'* classes. Openings in walls are represented by the *'IfcOpening'* class, related to the parent wall through the *'IfcRelVoidsElement'* relationship. The *'IfcAxis2Placement3D'* class defines the opening location, and its geometric representation is created using the swept solid technique with the *'IfcExtrudedAreaSolid'* class. Figure 6 illustrates the main IFC classes used for the geometric representation of walls, while Figure 7 depicts the main IFC classes used for the geometric representation of openings (Zbirovský, 2024).

3. Results

The process of converting a digital 3D scan into a 3D model was demonstrated on a slice of data from a real building. The Hotel Opatov in Prague was undergoing renovation and conversion to residential housing, and the building was scanned as part of the construction. The point cloud was cropped to one room from one floor (Figure 1). The preparation of the point cloud data involved noise removal and subsampling. Subsampling, which reduces the number of points to enhance workflow efficiency, was performed using the CloudCompare program. A minimum point distance mode was employed, effectively reducing the number of points from the original 5,008,195 to 245,544 points. This reduction was crucial for streamlining the subsequent processes. While the program has demonstrated its effectiveness with large data packages (65 million points), it was important to determine the limit at which segmentation becomes impossible due to excessive dilution of the point cloud. Therefore, the subsampling parameter was set to 0.1 meters to test this threshold. The following figures present visual representations of this study's findings. Figure 8 showcases the precision and clarity achieved through the developed methodologies by displaying the generated floor plan of the tested room in ArchiCAD 26. Figure 9 presents the program's output for the same room, visualized in ArchiCAD 26's 3D view. These visuals demonstrate the software's practical applicability and confirm its effectiveness in producing detailed and accurate BIM models.

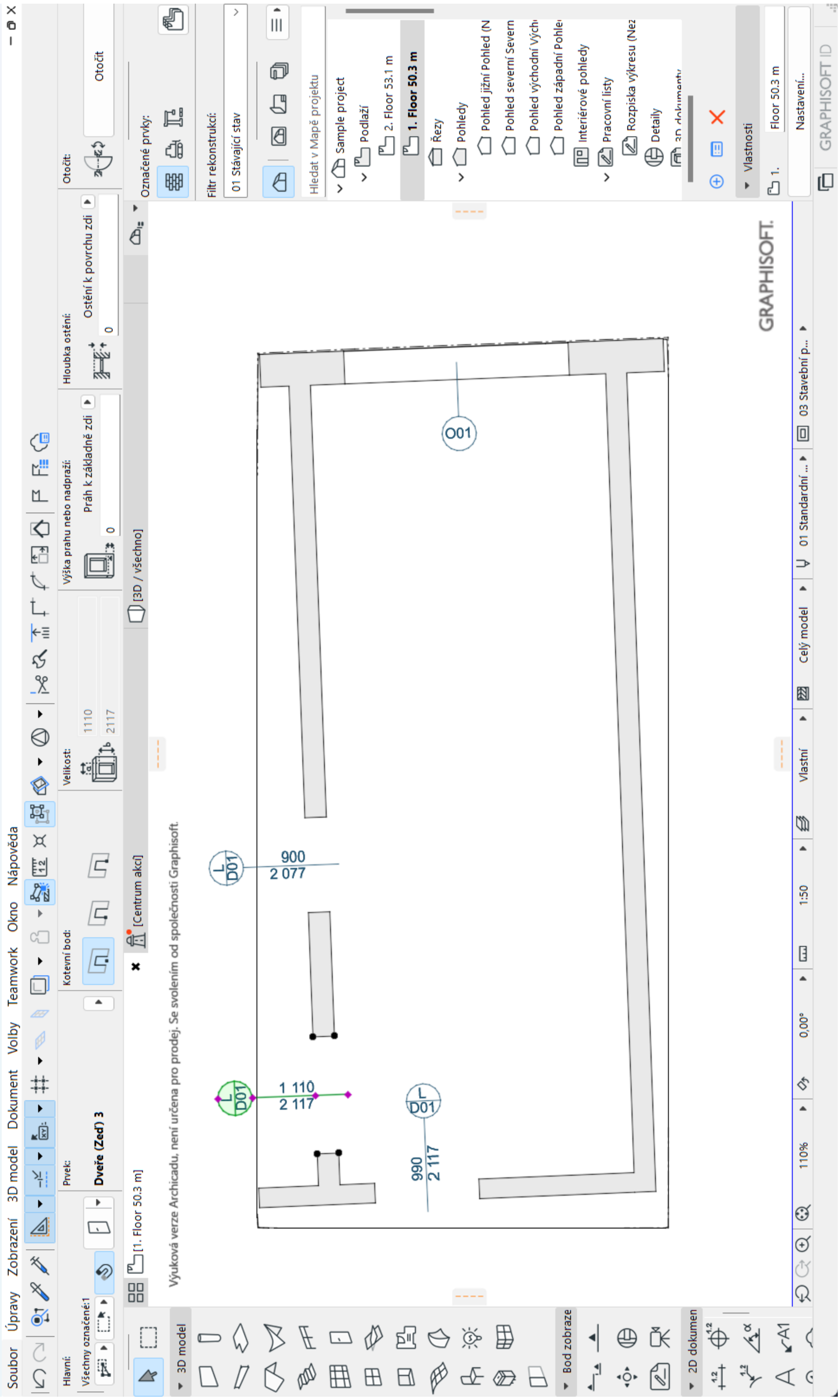

Figure 8: Generated floor plan of the testing room displayed in ArchiCAD 26 (Zbirovský, 2024).

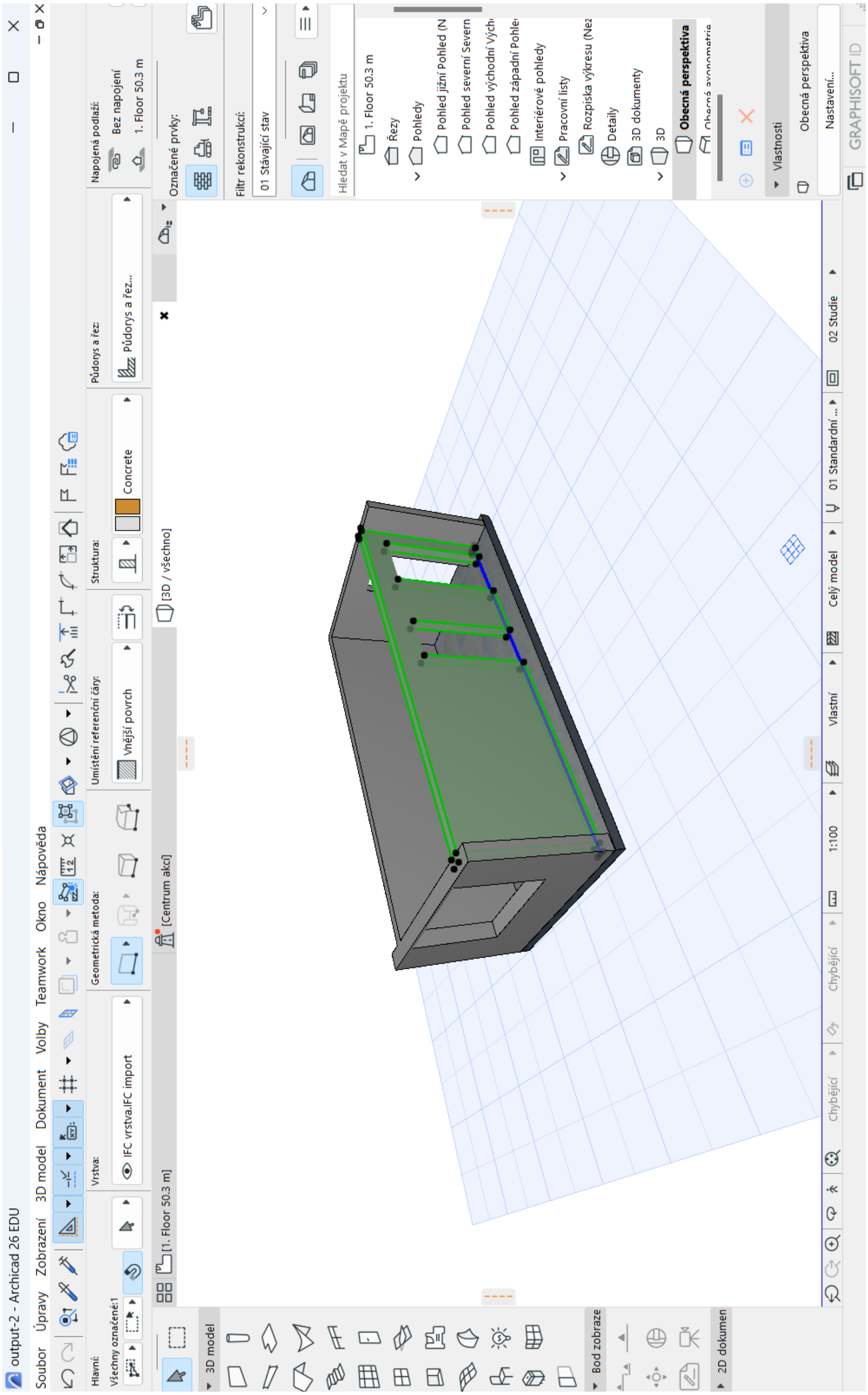

Figure 9: Program output of testing room displayed in ArchiCAD 26 (Zbirovský, 2024).

4. Conclusion

The presented software solution converts point cloud data into a 3D parametric BIM model. The solution includes an algorithm for the segmentation and classification of structural components: (i) Slabs, (ii) Walls, (iii) Door openings, and (iv) Window openings. The software's design is unique due to its complexity and ability to convert a multi-story, unorganized point cloud into a meaningful 3D building model. The software is based on the open-source programming language Python and other feature packages available for this tool. Complete source code is available at Github repository.

The program outputs data in the IFC file format, which can be opened in any OpenBIM modeling software. Leveraging the geometric primitives and complex shapes supported by the DesignTransferView_V1.0 schema, users can easily modify the resulting IFC files within their modeling software, ensuring seamless integration and collaboration.

In future work, it would be beneficial to implement a solution to the issue of walls that are only scanned from one side. This issue is particularly prevalent in the case of exterior walls when a digital scan is only taken from within the building, or when certain rooms are not accessible during the scanning process.

Acknowledgments

This work was funded by the European Union under the project ROBOPROX (no. CZ.02.01.01/00/22_008/0004590), by the European Union's Horizon Europe Framework Programme (call HORIZON-CL4-2021-TWIN-TRANSITION-01-11) under grant agreement No. 101058580, project RECONMATIC (Automated solutions for sustainable and circular construction and demolition waste management), and the Czech Technical University in Prague, grant agreement No. SGS24/003/OHK1/1T/11 (Application of state-of-the-art technologies for enhancing sustainability of the construction sector).

References

Bassier, M., & Vergauwen, M. (2020). Topology reconstruction of bim wall objects from point cloud data. *Remote Sensing*, 12(11). https://doi.org/10.3390/rs12111800

Chen, Z. (2022). From point cloud to as-built bim: Automated detection of openings in existing buildings. *Proceedings of 33. Forum Bauinformatik*.

IfcOpenShell. (2023). Ifcopenshell 0.7.0 documentation, python bindings, utility functions, and high-level api for ifcopenshell. Retrieved July 6, 2023, from www.pypi.org/project/ifcopenshell

Mahmoud, M., Chen, W., Yang, Y., & Li, Y. (2024). Automated bim generation for large-scale indoor complex environments based on deep learning. *Automation in Construction*, 162, 105376. https://doi.org/https://doi.org/10.1016/j.autcon.2024.105376

Martens, J., & Blankenbach, J. (2023). Vox2bim+-a fast and robust approach for automated indoor point cloud segmentation and building model generation. *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 91(4), 273–294. https://doi.org/https://doi.org/10.1007/s41064-023-00243-1

Mura, C., Mattausch, O., & Pajarola, R. (2016). Piecewise-planar reconstruction of multi-room interiors with arbitrary wall arrangements. *Computer Graphics Forum*, 35(7), 179–188. https://doi.org/https://doi.org/10.1111/cgf.13015

Nikoohemat, S., Peter, M., Oude Elberink, S., & Vosselman, G. (2018). Semantic interpretation of mobile laser scanner point clouds in indoor scenes using trajectories. *Remote Sensing*, 10(11). https://doi.org/10.3390/rs10111754

Ochmann, S., Vock, R., & Klein, R. (2019). Automatic reconstruction of fully volumetric 3d building models from oriented point clouds. *ISPRS journal of photogrammetry and remote sensing*, 151, 251–262. https://doi.org/https://doi.org/10.1016/j.isprsjprs.2019.03.017

Pintore, G., Mura, C., Ganovelli, F., Fuentes-Perez, L., Pajarola, R., & Gobbetti, E. (2020). State-of-the-art in automatic 3d reconstruction of structured indoor environments. *Computer Graphics Forum*, 39(2), 667–699. https://doi.org/https://doi.org/10.1111/cgf.14021

Tran, H., Khoshelham, K., & Kealy, A. (2019). Geometric comparison and quality evaluation of 3d models of indoor environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 149, 29–39. https://doi.org/https://doi.org/10.1016/j.isprsjprs.2019.01.012

Yang, F., Zhou, G., Su, F., Zuo, X., Tang, L., Liang, Y., Zhu, H., & Li, L. (2019). Automatic indoor reconstruction from point clouds in multi-room environments with curved walls. *Sensors*, 19(17), 3798. https://doi.org/https://doi.org/10.3390/s19173798

Zbirovský, S., & Nežerka, V. (2024). Transforming point cloud data into 3d bim model: A case study on slabs [Preprint]. *Acta Polytechnica CTU Proceedings*.

Zbirovský, S. (2024). *From point cloud to digital twin: Streamlining bim design automation* [Master's thesis, České vysoké učení technické v Praze.]. https://dspace.cvut.cz/handle/10467/113868