2025 European Conference on Computing in Construction

CIB W78 Conference on IT in Construction

Porto, Portugal
July 14-17, 2025

A HYBRID BOTTOM-UP TOP-DOWN AI METHOD FOR IMPROVED DIGITAL TWINNING OF THE BUILDINGS USING POINT CLOUD DATA AND RGB IMAGES

Mansour Mehranfar[1], Alexander Braun [1], and André Borrmann[1]
[1]Chair of Computing in Civil and Building Engineering, TUM Georg Nemetschek Institute
Technical University of Munich, Germany

Abstract

Digital twins have become transformative tools in design and operations, providing critical capabilities for real-time monitoring and management of building assets. However, creating high-quality digital building models required for the digital twinning of the built environment on a large scale remains challenging and requires significant human effort. This paper introduces an AI-based end-to-end automatic procedure for the creation of digital building models using point clouds and RGB images. The results demonstrate the effectiveness of the proposed method across multiple case studies, achieving an average accuracy of approximately 7 cm in estimating the parameters of the building's structural and opening elements.

Introduction

Digital Twinning of the Built Environment

Recent advances in information technology, particularly in Artificial Intelligence (AI) and Machine Learning (ML), have opened new opportunities for the Architecture, Engineering, and Construction (AEC) sectors.

One significant application of these technologies is the digitization of built environments by creating digital building models from remote sensing data, such as point clouds and RGB images. These digital models offer valuable capabilities for monitoring and managing building assets, enhancing the accuracy and flexibility of planning and maintenance processes, and improving transparency in decision-making.

Over the last decade, the development of digital twin (DT) concepts and their application across various fields, including industry, construction, and medicine, has led to their integration into diverse research areas (Sacks et al., 2020). Depending on the specific purpose, a DT model can encompass a wide range of information. Given its broad and versatile nature, the concept of a DT can be refined and scaled to provide a more precise definition tailored to its intended use (Brilakis et al., 2019).

In this paper, building digital twinning and the DT concept are defined as a purpose-driven semantic digital representation of building physical assets (Noichl et al., 2024). These representations can be regularly updated by transferring data between physical assets and their digital replicas and include 3D geometric models of building elements, which are further enriched with semantic information and topological relationships.

Background and Related Work

The creation of digital building models is fundamental to the effective implementation of DT for the assets of the built environment. The manual creation of digital building models is labor-intensive and time-consuming, requiring significant human effort and expertise. However, AI technologies offer promising advancements for developing automated digital building model creation methods, significantly reducing time and costs and making it more efficient and scalable (Pan et al., 2023).

Creating digital building models typically consists of two major steps: data capturing and data processing (Noichl et al., 2024). In the data capturing step, raw visual and spatial data such as RGB images and point clouds are collected using remote sensing technologies such as laser scanning and photogrammetry, efficiently acquiring geometric information of the built environment. Data processing involves interpreting raw point cloud data and images to extract meaningful, readable information for humans and machines, subsequently creating digital models of target objects. This typically involves segmentation, detection, and classification of different building elements to create a structured model through Bottom-Up, Top-Down, and AI-based techniques.

The **Bottom-Up** approach encompasses a range of techniques for segmenting point clouds, each with specific strengths and limitations. Common methods include Region Growing, Model-Based, and Edge-Based techniques, which start from individual data points and build up to form higher-level structures (Dimitrov and Golparvar-Fard, 2015).

In the **Top-Down** approach, the segmentation and classification are guided by predefined relationships among the elements, allowing for the identification and organization of various components based on their spatial and semantic attributes. The Top-Down approach typically requires a substantial amount of prior knowledge about the structure and characteristics of the data, necessitating the establishment of rules and reasoning frameworks to facilitate the segmentation process (Ochmann et al., 2019; Tran and

Khoshelham, 2020).

AI-driven approaches significantly enhance the ability to analyze and interpret point cloud data and images using semantic segmentation, object detection, and classification methods (Pan et al., 2022). In contrast to the often manual and rule-based segmentation processes used in Bottom-Up methods, AI can automate the identification and classification of features within point clouds and images (Tang et al., 2022). Despite their advantages, AI-based approaches require large annotated datasets for training, which can be challenging and time-consuming to obtain. Furthermore, AI methods may struggle with certain edge cases or rare scenarios that are underrepresented in training data, potentially leading to inaccuracies in segmentation or classification (Pan et al., 2022). Thus, the sole reliance on AI-based methods for data processing can disrupt semantic understanding and interpretation of topological relationships between elements, ultimately hindering the creation of high-quality digital building models with rich semantic detail and coherent geometry.

This research aims to introduce a hybrid Bottom-Up, Top-Down AI method for improved digital twinning of buildings. The goal is to use the strengths of all mentioned data processing approaches by combining domain engineering knowledge in building design and construction with AI capabilities in scene understanding to create digital building models with rich semantics and coherent geometry using point cloud data and RGB images.

Methodology

As shown in Figure 1, the proposed workflow for the automatic creation of digital building models, according to the schematic design at Level of Detail (LOD 200), consists of four major steps: (1) preprocessing, (2) semantic labeling, (3) creation of parametric building model, and (4) door and window detection. The following subsections provide details for each step.

Preprocessing

Effective preprocessing of input point cloud data is essential for optimizing subsequent analyses' quality and computational efficiency. Key steps in preprocessing include subsampling the point cloud to improve processing speed and filtering out noise and clutters.

Subsampling reduces the number of points in a point cloud while preserving the geometric structure and essential information, thereby reducing computational complexity. Common techniques include random sampling and grid-based sampling. Random sampling selects points at random, which is efficient, but may lead to uneven distribution and potential loss of critical features. Grid-based sampling, on the other hand, divides the point cloud into equal-sized cells with a dimension of d, selecting one point per cell to maintain uniformity and preserve fine details. Therefore, grid-based subsampling is chosen to enhance processing efficiency and ensure effectiveness.

Noise and outliers are outer points that deviate from the primary data distribution, commonly resulting from sources such as sensor inaccuracies, environmental variability, or transient objects. Noise and outliers points can be effectively removed using statistical or neighborhood-based analysis techniques. Among these, Connected Components Segmentation (CCS) is particularly effective; it applies a predefined distance threshold to identify clusters of connected points, treating each cluster as a distinct segment (Trevor et al., 2013). In indoor environments, noise and outliers frequently appear as small or isolated segments, facilitating their detection and removal from the dataset.

Semantic Labeling

Semantic labeling refers to assigning meaningful labels to points at various levels of contextual information, making the data more interpretable for processing by humans and machines.

Point Labeling

In this study, a pre-trained semantic segmentation model based on the PointTransformer network is employed to assign semantic labels to data at the point level (Zhao et al., 2021) (Figure). The PointTransformer network for semantic segmentation utilizes self-attention in combination with simple linear layers and a multi-layer perceptron (MLP) (Zhao et al., 2021). This model is specifically trained on the Stanford 3D Indoor Dataset (S3DIC), a widely recognized benchmark for 3D indoor point cloud processing tasks, which includes thirteen object classes such as *Ceiling, Floor, Wall, Clutter, and Furniture* (Armeni et al., 2016). The PointTransformer network was trained on areas 1-4 and 6, and its performance was then evaluated on the semantic segmentation of area 5. Table 1 presents the configuration and values of the hyperparameters used to train the network. These values were selected based on recommendations from prior research, determined by fine-tuning within specified ranges.

Table 1: Parameters of pre-trained semantic segmentation model using PointTransformer network.

Parameter	Value
Model:	
Input channels	6
voxel size	0.04
Max voxels	50000
Number of points per voxel	40960
Max epoch	512
Optimizer:	
Learning rate	0.01
Momentum	0.9
Weight decay	0.0001

Ceilings, floors, and walls are the main structural elements of buildings, playing essential roles in defining structural skeleton and spatial layout. Therefore, the primary objective of this step is to accurately separate points for Ceiling, Floor, and Wall elements. In this regard, the Point-

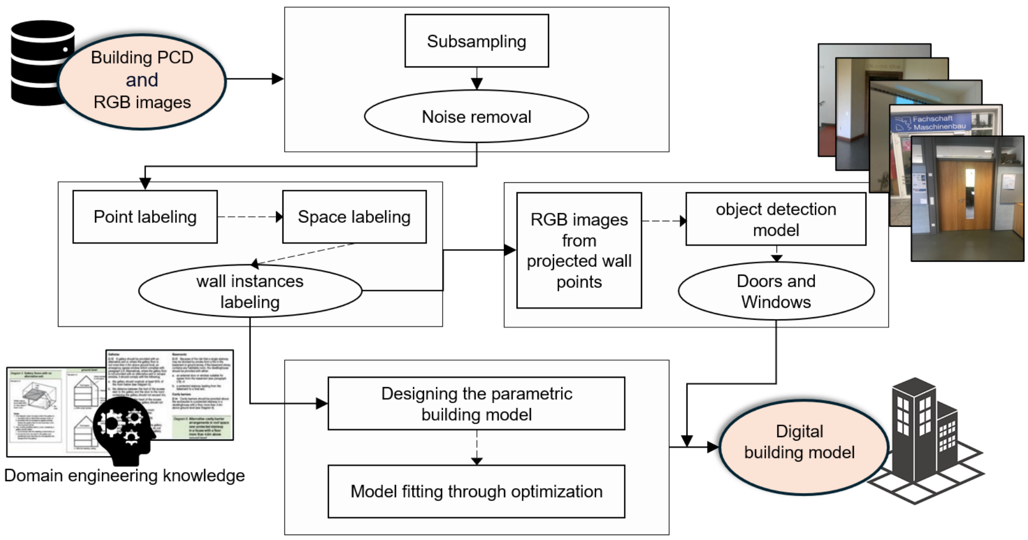


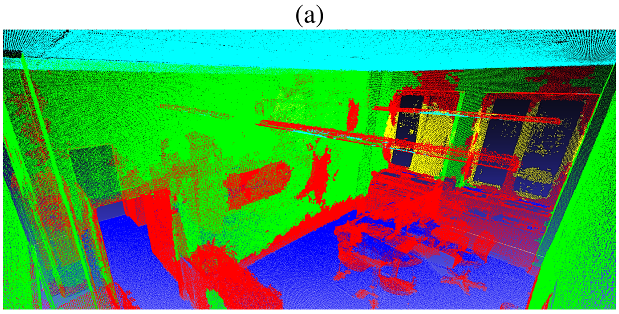
Figure 1: The proposed workflow for creating the digital building models using dense point cloud and RGB images.

Transformer model achieved an average accuracy of approximately 93% in segmenting main structural elements, highlighting its effectiveness in identifying the main structural elements within the complex and cluttered building point cloud.

Table 2: Semantic segmentation results on the S3DIS dataset, evaluated on Area 5 (Zhao et al., 2021).

PointTrasnformer:	
OA	90.8
mAcc	76.5
mIoU	70.4
Accuracy:	
ceiling	94.0
floor	98.5
wall	86.3
door	74.3
window	63.4
others	62.4

(a)



(b)

floor ceiling wall door window others

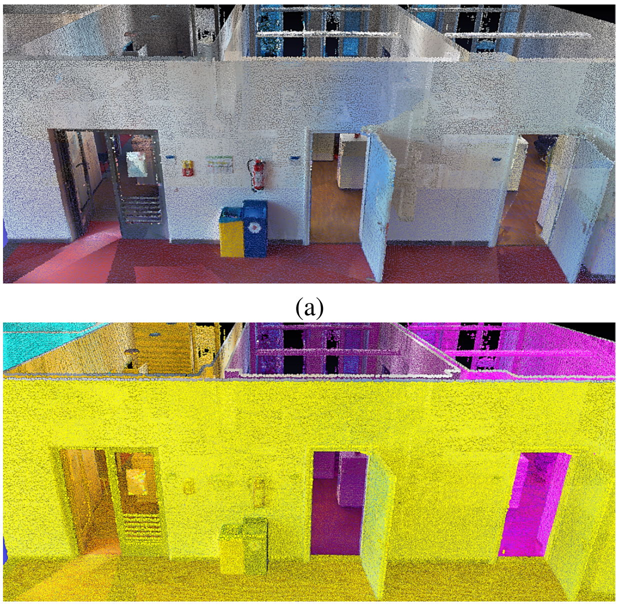
Figure 2: The result of point labeling using PointTransformer semantic segmentation model; (a) original point cloud of TUM Building 1 dataset, (b) the result of the point semantic labeling.

Space Labeling

Segmentation and geometric representation of individual rooms and spaces are essential for developing an accurate and detailed virtual model of a building layout and serve as a key reference to support effective planning and to optimize the functional use of architectural spaces.

A room or space is an enclosed area within a building, separated from other areas by walls and characterized by specific functional and structural traits. These characteristics may include various attributes such as size, shape, ceiling height, floor type, and other distinctive features unique to that particular space. This research employs a knowledge-based bottom-up approach to automatically label the data at the space level and segment individual wall instances within the point cloud.

For 3D space labeling, ceiling points at *th* distance from wall points are removed from the initial ceiling segment, resulting in remaining ceiling point clouds that form scattered segments farther from the walls. A density-based clustering algorithm (DBSCAN) is then applied to these remaining ceiling points to generate unique segments (Czerniawski et al., 2018). Finally, a hierarchical nearest neighbor method assigns the correct cluster labels to each 3D point in the point cloud (Figure 3). The appropriate value for *th* is determined based on the average thickness of walls commonly found in the specific type of buildings under examination.

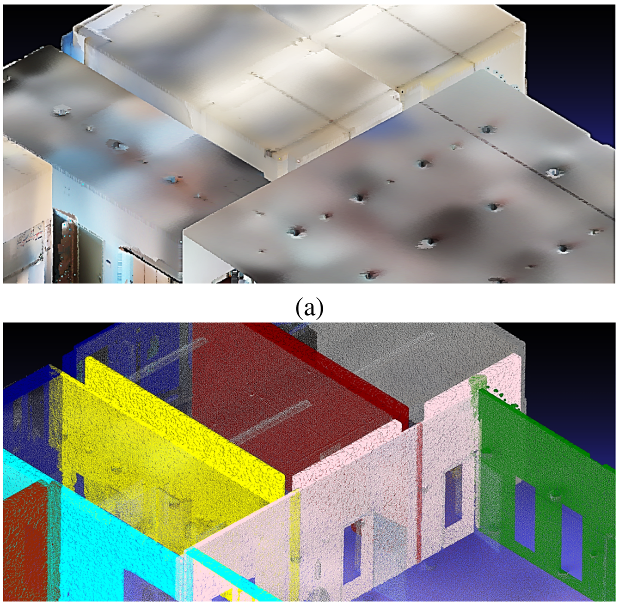


(a)

(b)

space (1) space (2) space (3) space (4) space (5)

Figure 3: 3D space parsing of the indoor point cloud; (a) the original point cloud of the TUM Building 1 dataset, (b) the result of the point labeling at the space level.



(a)

(b)

wall (1) wall (2) wall (3) wall (4) wall (5) wall (6)

Figure 4: 3D wall instance segmentation; (a) the original point cloud of the TUM Building 1 dataset, (b) the result of the point labeling at the wall instance level.

Wall Instances Labeling

The complexity of the indoor environment, along with the presence of noise and clutter, as well as the similarity in geometric features between walls and other furniture, can result in inaccuracies when labeling wall points and subsequent wall instances separation.

The proposed method uses a bottom-up, knowledge-based approach to detect wall footprints within each enclosed space and subsequently label individual 3D wall instances. Ceiling and wall elements share common boundaries within an enclosed space, and the boundary points of the ceiling can be leveraged to extract the wall footprints on the 2D X-Y plane. In scenarios with intersecting walls, changes in the principal component analysis (PCA) parameters of the wall instances can reveal breakpoints or abrupt shifts. These serve as endpoints for each wall instance, indicating changes in curvature.

In this method, the Mean Shift and Alpha-shape algorithms are initially employed to extract the ceiling's boundary points per space. Subsequently, the PCA coefficient values are calculated for each boundary point p by considering its k neighbor points and determining the covariance matrix, denoted as c, through Equation (1):

$$c = \frac{1}{k}\sum_{i=1}^{n}(p_{\mathrm{i}} - \bar{p}).(p_{\mathrm{i}} - \bar{p})^{T} \tag{1}$$

where k is the number of neighboring points, p_i and p also refer to the coordinates of the boundary points being considered.

The boundary points are subsequently grouped into different groups depending on their orientation based on their PCA coefficient values. These groups can be denoted as vertical, horizontal, and inclined classes. To group the boundary points that correspond to the same wall instances, the DBSCAN clustering algorithm is employed. Finally, the points corresponding to each wall within the 3D space are extracted from the original point cloud by considering the buffer b around the wall instances (Figure 4). The method can effectively separate and group wall instances based on their orientation.

Creation of parametric building model

To create the digital building model, information from the 3D space parsing and wall instance extraction steps is utilized in a bottom-up approach. This process generates an initial floor plan mask using a plane-plane intersection method, which is subsequently extruded into a 3D volumetric model (Figure 5). However, due to errors in the plane-fitting process and the presence of noise and clutter, the resulting model may exhibit inaccuracies in wall locations as well as width and height estimations.

In this regard, a top-down approach is used to address the issue. First, domain engineering knowledge in the fields of BIM and building design is used to establish geometrical-mathematical rules and constraints, which are applied as internal relationships between system elements. These rules define the interactions between elements (e.g., walls, slabs, etc.) and specify the degrees of freedom and the range of permissible changes for the parameters of these elements. In the proposed parametric digital building model, each wall instance can adopt one of three possible orien-

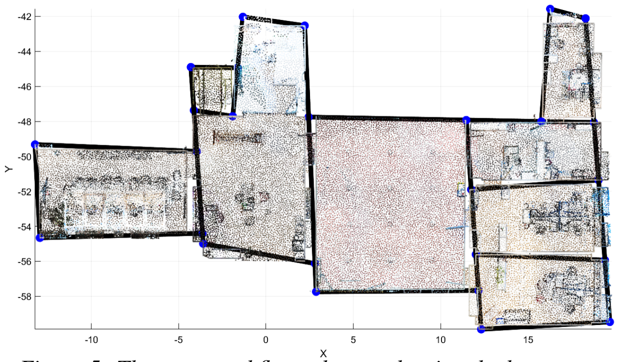


Figure 5: The generated floor plan mask using the bottom-up plane-plane intersection method.

tations: horizontal, vertical, or inclined. The defined rules and constraints specify the interactions between each wall instance and its connected walls.

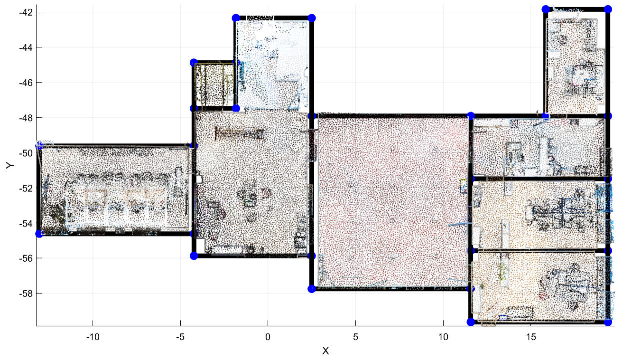
Despite maintaining consistent semantic topology, the designed parametric model may exhibit low geometric accuracy in element property values and their positions within the environment. To address this, the model is refined in the next step by fitting it to the point cloud using the Particle Swarm Optimization (PSO) algorithm (Kennedy and Eberhart, 1995), which extracts optimal values for the model's parameters (Figure 6b). During the model fitting process, walls are represented as boxes with dimensions (width, length, height) varying to accommodate different values.

In the global optimization problem, the overall Points-to-Model distance serves as the objective function to create accurate geometric models by optimizing dimensional properties and locations. This involves calculating the distances between points and the surfaces of the digital model. While utilizing global optimization for parametric model fitting may increase computational complexity, it ensures logical consistency throughout the entire model reconstruction process. Figure 6c illustrates the best objective function values over different iterations during the optimization process for fitting the designed parametric model to the point cloud using the PSO algorithm.

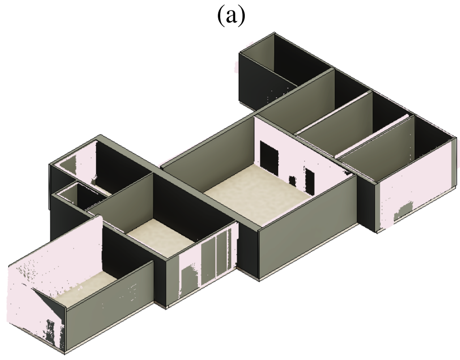
Door and window detection

Despite the high performance of the PointTransformer network for semantic segmentation of main structural elements, the separation of door and window elements results are suboptimal and are always associated with noise and over-segmentation problems. This issue arises from factors such as sparse points caused by glass reflections and the similarity of these elements' geometrical and spectral features to other structural components and results in an inaccurate estimation of the location and dimensional parameters of door and window instances within the built environment.

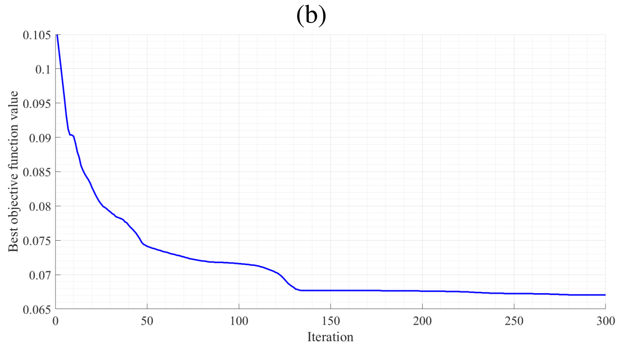
This research proposes a novel approach for improving the detection and digital modeling of door and window elements in indoor environments using YOLOv8 object detection network (Jocher et al., 2023).



(a)



(b)



(c)

Figure 6: Fitting the parameterized digital building model to point cloud data using PSO optimization algorithm; (a) the floor plan mask after optimization, (b) digital building model with optimum parameter values, (c) the trend of best objective function values over iterations during the optimization.

In this process, points belonging to one of the two wall surfaces (within a 1 m distance around each wall instance) are first projected onto the X-Z and Y-Z planes. Subsequently, a gridding and sampling method with dimensions of d is applied to convert the point cloud representation into an RGB image. The spectral values of the image pixels are estimated by calculating the average spectral values of the points within each grid cell.

Comprehensive image datasets consisting of two distinct sub-categories are collected to train the YOLOv8 object detection network for door and window detection (Figure 7). The first subset includes 214 RGB images captured from various buildings at the Technical University of Munich (TUM), featuring doors and windows in different states—open, semi-open, and closed—with a variety of materials and appearances. Additionally, 89 images generated from projected wall points of the TUM point cloud datasets are included to further diversify the training data. The bounding boxes for door and window elements were carefully annotated within the images. The annotated dataset is then divided into training and validation subsets in an 80:20 ratio and the object detection model is trained using the hyperparameters specified in Table 3.

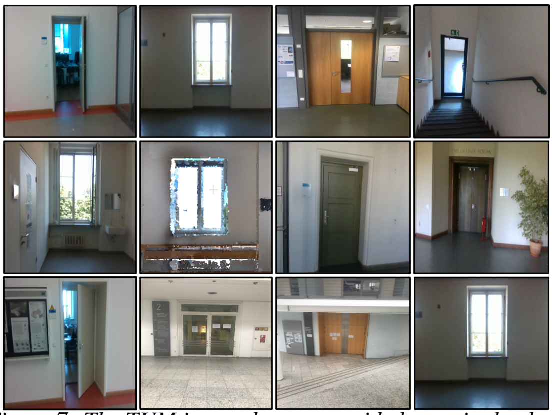


Figure 7: The TUM image dataset provided to train the door and window detection network.

Table 3: The hyperparameter values employed in training the object detection network using the YOLOv8 architecture.

Parameter	Value
Image size	640
Batch size	8
Epoch	150
Learning rate	0.001
Solver	Adam

To evaluate the network's performance in the training phase, the annotated element instances in the images are compared with the detected instances using standard metrics Precision, Recall, and mean average precision (mAP), through Equations 2-4:

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$mAP = \frac{1}{classes} \Sigma_{c \in classes} \frac{|TP_c|}{|FP_c|+|TP_c|} \tag{4}$$

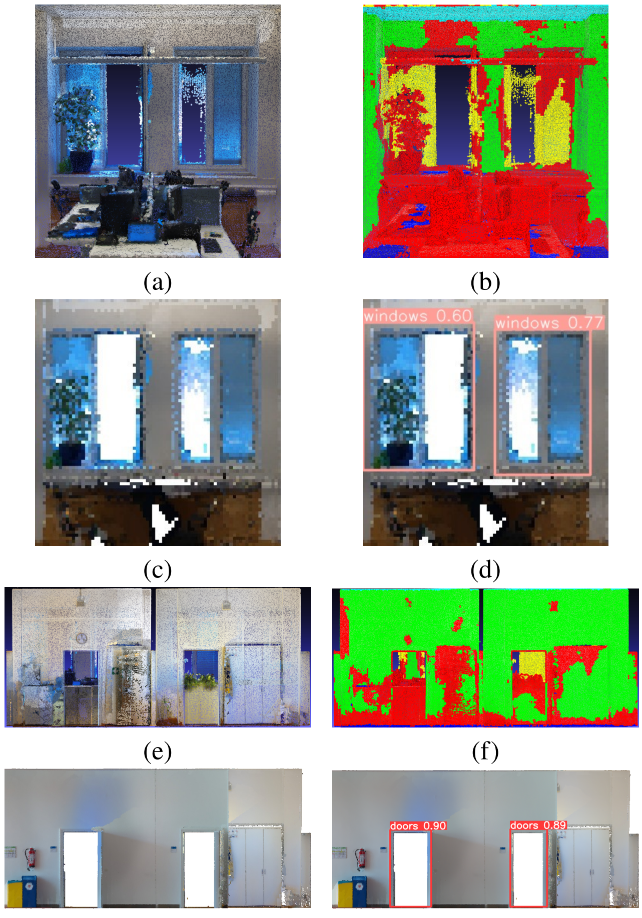
Where TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively. In this regard, the mAP of about 95% in the learning process highlights the effectiveness of the utilized object detection network in the detection of door and window elements (Tables 4-5).

Creation of digital door and window model

After detecting the door and window elements in the images, their bounding box coordinates are projected back from 2D to the 3D space of the wall point cloud through reverse mapping. However, inaccuracies in projecting wall points to images and detecting exact door and window pixels using the object detection network can lead to errors in estimating the parameters of these elements.

Table 4: Accuracy evaluation of trained network for door and window detection, mAP (Mean average precision at IoU thresholds of 50 and 50-95).

Class	Precision	Recall	mAP(50)	mAP(50-95)
Doors	0.94	0.86	0.95	0.77
Windows	0.93	1.00	0.95	0.69
All	0.94	0.93	0.95	0.73



(a) (b) (c) (d) (e) (f) (g) (h)

floor ceiling wall door window others

Table 5: The result of door and window detection on the projected wall points images; (a) and (e) original point cloud, (b) and (f) the result of the point cloud semantic segmentation using PointTransformer model, (c) and (g) the projected wall instance points, (d) and (h) the result of door and window detection using YOLOv8 object detection model.

A comprehensive library of door and window families tailored for BIM applications is compiled to enhance accuracy and improve the integrity of element representation. For each detected door and window instance, the closest model from the library is selected based on dimensional parameters, such as width and height. The chosen model is then used to update the detected elements, replacing their initial primitive dimensions with accurate dimensional parameters (Figure 8).

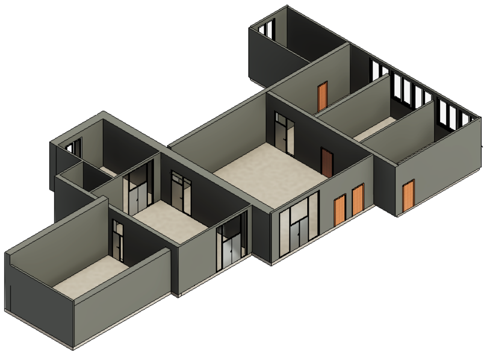


Figure 8: Parameterized digital building model, with the appropriate representation of doors and windows in the reconstructed digital model.

Result

Case study

To evaluate the performance of the proposed method, four distinct indoor point cloud datasets from different buildings on the TUM city campus (https://doi.org/10.14459/2024mp1742891) and the NavVis company office are used (https://www.navvis.com). These building datasets include both Manhattan and non-Manhattan designed and are primarily utilized for educational and research purposes featuring various spaces, such as offices, libraries, meeting rooms, and hallways.

Implementation

The proposed pipeline is implemented in Python on a research computer (11th Gen Intel(R) Core(TM) i7-1165G7, with 16.0 GB1053 memory) and is tested on each individual point cloud dataset. Table 6 provides an overview of the key parameters used for creating the digital models. The parameter values were determined through fine-tuning within a specified experimental range and are applicable to other building datasets in real-world scenarios.

Table 6: Overview of the essential parameters employed during the creation of digital model using the proposed method.

Parameter	Value
Preprocessing:	
1. Grid-based subsampling distance	0.05 m
2. Distance *th* for noise removal	0.25 m
Semantic labeling:	
3. Ceiling-Wall distance threshold (space parsing)	0.30 m
4. Max_Dist for the DBSCAN clustering	0.30 m
5. Number of NN for calculating PCA coefficients	50
Door and window detection:	
6. Grid size for converting points into images	0.05 m

Experimental results of digital model representation

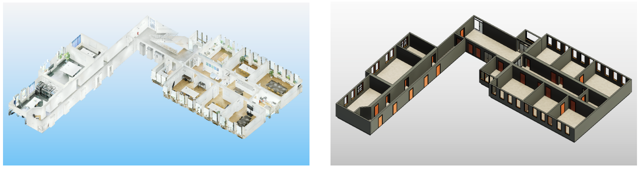
To assess the performance of the proposed method for the automatic creation of digital building models, the parameter values for walls, doors, and windows in the reference digital models are compared with the corresponding values in the reconstructed digital model. In this process, corresponding elements in both the reference and reconstructed models are identified using their coordinates and a buffer with specific dimensions of 10 cm (Table 7). The accuracy of the reconstruction for each model is then evaluated by calculating Precision and Recall values based on Equations 2-3. The overall mean accuracy of approximately 7 cm in estimating models' parameters highlights the performance and effectiveness of the proposed method for the automatic creation of digital models for building structures in the built environment.

Table 7: Accuracy evaluation of digital model reconstruction (the values for the reported parameters in the table are all in cm).

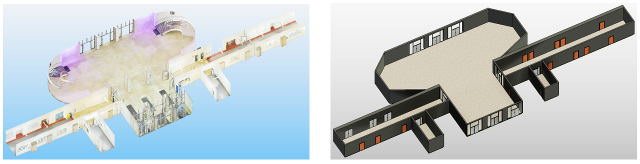
Dataset	Data (1)	Data (2)	Data (3)	Data (4)
Wall:				
precision	0.96	0.68	0.95	0.96
recall	0.85	0.81	0.86	0.96
δ location	4.9	6.2	5.3	6.8
δ dimension	4.8	5.7	5.7	6.7
Door and Window:				
precision	0.97	1.00	0.95	1.00
recall	0.98	0.84	0.56	0.93
δ location	8.7	7.1	6.4	6.6
δ dimension	9.1	8.7	9.3	9.8

Conclusion

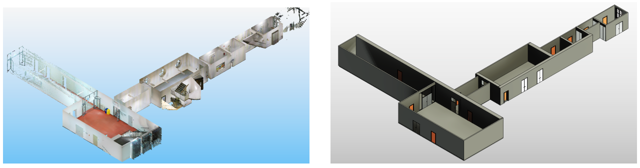
This research proposes a novel hybrid bottom-up and top-down AI method for improved digital twinning of buildings using point clouds and RGB images. In contrast to the developed bottom-up approach, the proposed method integrates existing knowledge from building design and construction with AI capabilities in scene understanding to improve the geometric consistency and the semantic relationships between elements in the resulting digital building models. The proposed parametric model fitting method, using global optimization, overcomes the challenges posed by noise and data gaps, minimizing their impact on the overall model reconstruction process. The proposed approach for detecting door and window elements from projected wall points using AI object detection has shown superiority over the bottom-up semantic segmentation method, allowing for detecting doors and windows with various types and appearances within the indoor environment. Despite careful consideration, the proposed method faces challenges in accurately separating and modeling curved wall instances. Additionally, the method for detecting door and window elements can only identify instances with rectangular appearances and lacks the ability to detect doors and windows with round geometric shapes. Future research can be carried out to address these challenges and enhance the development and semantic information of the resulting models by incorporating digital representations of other structural and architectural elements, such as staircases, columns, and more.



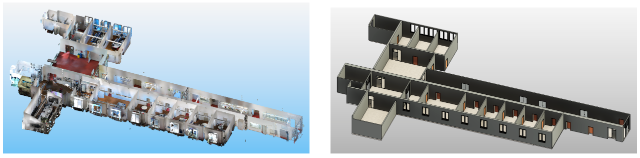
Data (1) - NavVis office building



Data (2) - TUM main entrance



Data (3) - TUM building (1) - Floor (2)



Data (4) - TUM building (1) - CMS chair

Table 8: Overview of the datasets and corresponding reconstructed digital building models.

Acknowledgments

The research presented has been performed in the frame of the project AI4TWINNING (Artificial Intelligence for the automated creation of multi-scale digital twins of the built world) funded by the TUM Georg Nemetschek Institute Artificial Intelligence for the Built World (GNI).

References

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., and Savarese, S. (2016). 3D Semantic Parsing of Large-Scale Indoor Spaces. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 1534–1543, Las Vegas, NV, USA. IEEE.

Brilakis, I., Pan, Y., Borrmann, A., Mayer, H.-G., Rhein, F., Vos, C., Pettinato, E., and Wagner, S. (2019). Built environment digital twining. International Workshop on Built Environment Digital Twinning presented by Technical University of Munich.

Czerniawski, T., Sankaran, B., Nahangi, M., Haas, C., and Leite, F. (2018). 6D DBSCAN-based segmentation of building point clouds for planar object classification. Automation in Construction, 88:pp. 44–58.

Dimitrov, A. and Golparvar-Fard, M. (2015). Segmentation of building point cloud models including detailed architectural/structural features and MEP systems. Automation in Construction, 51:pp. 32–45.

Jocher, G., Chaurasia, A., and Qiu, J. (2023). Ultralytics YOLO.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In Proceedings of ICNN'95 - International Conference on Neural Networks, volume 4, pages 1942–1948 vol.4.

Noichl, F., Collins, F. C., Pan, Y., Mafipour, M., Mehranfar, M., and Borrmann, A. (2024). From purpose to technology: A requirements-driven approach to designing digital twin implementations. In Jebelli, H., Asadi, Somayeh andMutis, I., Liu, R., and Cheng, J., editors, Digital Twins in Construction and the Built Environment, pages 91–130. American Society of Civil Engineers (ASCE).

Ochmann, S., Vock, R., and Klein, R. (2019). Automatic reconstruction of fully volumetric 3D building models from oriented point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 151:pp. 251–262.

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

Pan, Y., Braun, A., Borrmann, A., and Brilakis, I. (2023). 3D deep-learning-enhanced void-growing approach in creating geometric digital twins of buildings. Smart Infrastructure and Construction, 176(1):pp. 24–40.

Sacks, R., Brilakis, I., Pikas, E., Xie, H. S., and Girolami, M. (2020). Construction with digital twin information systems. Data-Centric Engineering, 1:14.

Tang, S., Li, X., Zheng, X., Wu, B., Wang, W., and Zhang, Y. (2022). BIM generation from 3D point clouds by combining 3D deep learning and improved morphological approach. Automation in Construction, 141:104422.

Tran, H. and Khoshelham, K. (2020). Procedural Reconstruction of 3D Indoor Models from Lidar Data Using Reversible Jump Markov Chain Monte Carlo. Remote Sensing, 12(5):838.

Trevor, A. J., Gedikli, S., Rusu, R. B., and Christensen, H. I. (2013). Efficient Organized Point Cloud Segmentation with Connected Components. In Proceedings of the semantic perception and mapping exploration, pages 1–6, Karlsruhe, Germany. last access: March 09, 2024.

Zhao, H., Jiang, L., Jia, J., Torr, P. H., and Koltun, V. (2021). Point Transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 16259–16268, Montreal, QC, Canada. IEEE.