ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XI-2-2026
XXV ISPRS Congress 2026 “From Imagery to Understanding”, Commission II, 4–11 July 2026, Toronto, Canada

Point2WSS: Reconstructing LoD2 Buildings from Aerial LiDAR Data using Multimodal Learning and Weighted Straight Skeleton

Pierre-Loïc Queffélec[1,2], Nicolas Trouvé[1], Stéphane Roussel[1], Teng Wu[2], Bruno Vallet[2]

[1] DEMR, ONERA, Université Paris Saclay, F-91123 Palaiseau, France - (pierre-loic.queffelec, nicolas.trouve, stephane.roussel)@onera.fr
[2] Univ Gustave Eiffel, Géodata Paris, IGN, LASTIG, F-77454 Marne-la-Vallée, France - (bruno.vallet, teng.wu)@ign.fr

Keywords: LoD2 Building Reconstruction, Multimodal Learning, Aerial LiDAR, Radar Simulation, Parametric Building Model

Abstract

In this paper, a method exploiting aerial LiDAR point clouds to build realistic building meshes suitable for electromagnetic simulation is proposed. One of the main challenges lies in reconstructing regularized building meshes with low polygonal density. Optimization-based methods, commonly used for building reconstruction from point clouds, are highly data-driven, making the quality of results dependent on the quality of input data. Aerial LiDAR scans can be incomplete or sparse, for instance due to occlusion. A novel LoD2 buildings reconstruction method based on deep learning is proposed, assuming that deep learning methods are more robust to incomplete or sparse data than optimization-based methods. A parametric building model is introduced, based on the Weighted Straight Skeleton algorithm, which generates realistic roofs from a building footprint and an associated set of slopes, and subsequently extrudes the roof to the specified building height. This parametric approach guarantees that a given set of parameters (height, footprint and slopes) produces a regularized building mesh with low polygonal density. A multimodal model, named Point2WSS, was trained to recover the variable number of building’s continuous parameters from its corresponding point cloud. This approach enables the generation of realistic building meshes suitable for electromagnetic simulation, if the predicted parameters accurately approximate real-world values. All the code and datasets used in this paper are available at : https://github.com/KWIKERRR/point2wss.

1. Introduction

Radar sensors have been widely adopted in both civilian and military industries, ranging from long-distance surveillance to vehicle speed monitoring, thanks to their all-weather active imaging capability. In the development and optimization of radar sensors, simulation tools have proven to be an efficient and reliable means for design and performance evaluation, such as EMPRISE (Trouvé et al., 2024) a radar simulator developed in ONERA. However, these simulators rely on the availability of realistic 3D scenes. Manually creating such scenes can be extremely time-consuming. Therefore, several studies have focused on automatically generating scenes from 3D acquisitions, such as LiDAR scans or multi view stereo. As part of the national HD LiDAR* program, the French Mapping Agency (IGN) produces and distributes a 3D mapping of the entire French territory in the form of point clouds sampled at 10 points per m². The underlying objective of this work is to exploit such point clouds to generate realistic 3D scenes designed for the specific requirements of radar and electromagnetic simulation. In particular, one of the main challenges lies in reconstructing building meshes that:

- Are regularized: preserving orthogonality, parallelism and collinearity is crucial for ensuring realistic radar interactions, particularly due to double-bounce effects. More generally, the aim is to produce a clean, manifold mesh without artifacts;
- Have low polygonal density: an excessive number of faces significantly increases the computational cost of radar simulations.

* https://geoservices.ign.fr/lidarhd

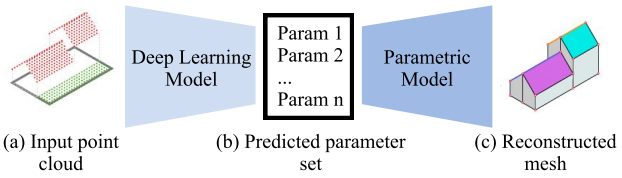


Figure 1. Proposed reconstruction method.

Current approaches (Bauchet et al., 2024, Peters et al., 2022, Huang et al., 2022), based on optimization methods, are highly data-driven, which makes them difficult to apply to sparse or incomplete data, such as HD LiDAR point clouds, where occlusion effects during acquisition often result in missing information.

This work presents a novel method, as illustrated in Figure 1, that combines a parametric building model with deep learning methods to generate 3D building meshes suitable for radar simulation from HD LiDAR point clouds. This method is based on two assumptions: (i) deep learning techniques are robust to incomplete and sparse data, and (ii) a sequence of parameterizable construction rules ensures a regularized building mesh with a low polygonal density. Another idea is that parameter prediction is a simpler task than directly predicting building meshes, introduced in (Liu et al., 2024, Li et al., 2022, Liu et al., 2021). A multimodal model, named Point2WSS, was trained to recover a variable number of building’s continuous parameters from its corresponding point cloud. These parameters serve as input of a parametric construction method based on the Weighted Straight Skeleton (WSS) algorithm (Held and Palfrader, 2017), allowing the generation of a wide variety of realistic roofs from a set of slopes, particularly adapted for residential buildings. If the model accurately predicts the construction

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-331-2026 | © Author(s) 2026. CC BY 4.0 License.

parameters, it allows the reconstruction of a realistic LoD2.0[†] (Biljecki et al., 2016b) building mesh suitable for radar simulation, making this approach highly promising.

This work first presents related research on building reconstruction from point clouds, highlighting the advantages and limitations of current approaches in Section 2. After introducing in Section 3 the WSS parametric building reconstruction method used in this study, the architecture of the multimodal model, Point2WSS, is described in Section 4. The training setup and results are then presented in Section 5, followed by quantitative and qualitative evaluations in comparison with existing methods. Finally, conclusions and perspectives are outlined in Section 6.

The key contributions of this work are:

- The development of a new method for building reconstruction from point clouds, integrating a parametric building model with multimodal learning;
- A multimodal model architecture capable of regressing a variable number of continuous outputs;
- A pipeline for generating point clouds annotated with construction parameters, facilitating the creation of large-scale synthetic datasets on demand;
- A flexible parametric building model enabling the reconstruction of diverse building geometries.

2. Related Works

2.1 Optimization-based Methods

Optimization-based methods are historically used for reconstructing buildings from point clouds and widely explore in several reviews (Xu and Stilla, 2021, Wang et al., 2018). These methods can be divided into two main categories: bottom-up and top-down, referring to their underlying philosophy — either constructing the building by combining primitives extracted from raw data toward the whole structure, or inversely fitting a global shape onto the input data.

Bottom-Up methods. They rely on primitives detection such as planes, lines or points. Directly detecting planar parts from point clouds is typically achieved using RANSAC (Schnabel et al., 2007), region-growing algorithms (Lafarge and Mallet, 2012), or Hough transforms (Overby et al., 2004). Recently, City3D (Huang et al., 2022) combines footprint extrusion, detected from a heightmap, with roof generation based on planar surfaces extracted from the point cloud. Other approaches, including SimpliCity (Bauchet et al., 2024) and Roofer (Peters et al., 2022), first extract a 2D polygonal partition of the roof structure, which is then extruded into 3D to reconstruct the complete geometry. These methods have shown their ability to handle various polygonal partitions but often produce overly-complex meshes. Furthermore, they are highly data-driven, making the quality of the results dependent on the input data and their optimization processes often rely on hyperparameter tuning.

Top-Down methods. Model-driven methods perform building reconstruction by matching a predefined library of buildings or building parts to the point cloud (Nys et al., 2020, Kada and McKinley, 2009). These methods struggle to produce buildings that cannot be decomposed precisely using the predefined library. Other methods focus on simplifying naive mesh constructed from the input point cloud (Boulch and Marlet, 2022, Li and Nan, 2021, Salinas et al., 2015).

2.2 Point Cloud–Conditioned Mesh Generation

With the increase in computational resources and the development of large language models, some research has focused on large-scale autoregressive models for mesh generation conditioned by an other modality such as images, NeRFs and point clouds. Some approaches process meshes as sequences of vertices and, similar to how language models generate text, produce them vertex by vertex. However, initial methods (Chen et al., 2024a, Nash et al., 2020) still struggle to generate complex meshes due to inefficient and highly redundant tokenization methods. Recent advances (Chen et al., 2024b, Tang et al., 2024) have focused on improving the efficiency of mesh tokenization by maximizing edge sharing between adjacent faces. More applied methods, including PC2WF (Liu et al., 2021), Point2Roof (Li et al., 2022) and Point2Building (Liu et al., 2024) focus on producing collections of building vertices and edges from an input point cloud. These approaches show promising performance for 3D mesh generation but suffer to produce regularized meshes.

2.3 Point Cloud–Conditioned CAD Reconstruction

Human designers traditionally use Computer-Aided Design (CAD) to construct diverse and complex 3D models, from chairs to airplanes. Human designers iteratively refined their model, following a top-down design approach, progressing from the general structure to the finer details. Naturally, some works focus on predicting a sequence of CAD instructions from a point cloud, making the reconstruction process straightforward. DeepCAD (Wu et al., 2021) learns a joint representation of CAD instructions and point clouds via an autoencoder. This allows point clouds to be embedded within the same latent space as CAD instructions, from which the decoder reconstructs the corresponding CAD instructions. A multimodal diffusion method, where noise is scheduled following top-down design process of human designers, to reconstruct masked CAD instructions is proposed by Draw Step by Step (Ma et al., 2024). The underlying intuition is that following the strategy of human designers can help to better reconstruct CAD instructions from point clouds. However, not all generated instructions can produce topologically valid shapes, and the learning process is constrained by the inherent ambiguity that a single 3D model can be generated from multiple distinct instruction sequences. Other methods, including SECAD-Net (Li et al., 2023) and Point2Cyl (Uy et al., 2022), propose to predict extrusion parameters of cylinders but might struggle to generate valid CAD instructions.

2.4 Inverse Procedural Generation

Procedural Building Generation (PBG), widely explored in (Kutzias and von Mammen, 2023), referred to methods that automatically create complex building trough L-systems (Parish and Müller, 2001), grammar (Müller et al., 2006) or parametric configurations (Biljecki et al., 2016a). Only a few works, such as (Zeng et al., 2018), focus on inverse procedural buildings generation – which consists of predicting automatically input parameters of a procedural building generation method –

[†] https://3d.bk.tudelft.nl/lod

highlighting the difficulty to automatically reverse engineered PBG methods. A recent approach (Župan et al., 2023) procedurally generates buildings from open-source data. However, despite the large amount of information collected from various sources, many buildings still require manual editing to accurately match reality.

3. WSS Parametric Building Model

This work introduces a parametric building generation method designed to be both straightforward, with a limited set of parameters, and flexible, enabling the modeling of a wide range of building types. This method is built on top of the Weighted Straight Skeleton (WSS) algorithm, which enable the reconstruction of realistic roofs from a footprint and a set of weights (Held and Palfrader, 2017) defining the slopes of each roof parts as illustrated in Figure 2. The generated roof is then extruded by growing roof parts with the defined slopes from (horizontal) gutters placed at the building’s height.

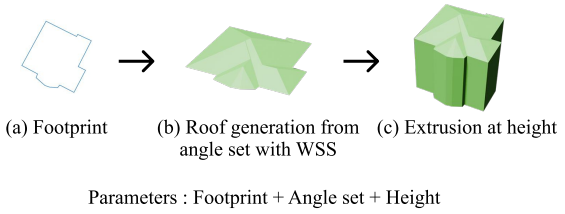


Figure 2. Weighted Straight Skeleton (WSS) reconstruction.

This method allows the generation of LoD2.0 buildings, even with complex structures and roof configurations because the topology is automatically created from the geometry (footprint and slopes). Furthermore, the WSS algorithm ensures that the generated meshes are regularized and have a low polygonal density. To make parameters more human-readable, slopes are converted to angles, each defining the angle between the horizontal plane and the corresponding roof part. The input parameter set of this method is composed of a height, a footprint and an angle set, as illustrated in Figure 2. However, this method is not able to reconstruct buildings with discontinuous heights or roof geometries that cannot be captured using a single angle per footprint edge. These limitations could be overcome by representing such buildings as combinations of multiple buildings that are generable with this method, but this approach is beyond the scope of the present work.

The input parameter set has variable length, since the number of angles and the footprint size are determined by the building structure. This property imposes constraints on the design of the deep learning model, as conventional regression models cannot handle variable-length outputs.

4. Point2WSS

Point2WSS was designed according to two main constraints, (1) the model should take as input both a point cloud and its corresponding footprint. Since the footprint is commonly used to extract a point cloud of a given building, it is assumed that footprint is known or can be inferred. This allow associating an angle with each edge of the footprint. (2) The model should be able to predict a variable number of continuous outputs, as the number of angle depends on the geometry of the footprint. Considering these constraints, a multimodal architecture appears to be an effective choice.

4.1 Overall Architecture

Multimodal models were originally developed in vision-and-language representation learning for tasks such as visual question answering and image-text retrieval. They are designed to reconstruct a masked signal of one modality with the help from another modality. By representing sequences of tokens for each modality, these models can naturally handle variable-length inputs and outputs, enabling the prediction of sequences whose length depends on the underlying data. (Kwon et al., 2023) proposes a vision-and-language model that jointly performs masked vision and language modeling, motivated by the nature of image-text paired data that both of the image and the text convey almost the same information but in different formats. As point clouds and parameter sets serve as complete but distinct representations of a 3D building mesh, their overall architecture is adopted in this work, with minor modifications to accommodate the specific requirements of point clouds and parameter sets. The overall idea is to reconstruct the parameter set token sequence, where each parameter value had been masked, from the point cloud token sequence, allowing the prediction of a variable number of continuous output.

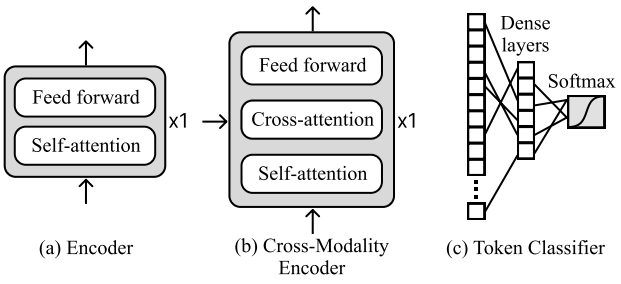


Figure 3. Point2WSS components.

The model consists of two main branches: (i) a point cloud reconstruction branch and (ii) a parameter set reconstruction branch, as illustrated in Figure 4. Both branches follow the same overall architecture, comprising an encoder, a cross-modality encoder, and a token classifier, as illustrated in Figure 3, and are trained together at each training step.

4.2 Point Cloud Reconstruction

This branch is designed to ensure a good local and global understanding of the input point cloud via a masking strategy. Moreover, the cross-attention with the parameter set helps accurately reconstruct the point cloud, enhancing this understanding. The point cloud reconstruction follows the same strategy as Point-BERT (Yu et al., 2022).

Tokenization. A tokenizer first learns a vocabulary of point cloud patches. The input point cloud is segmented, into 64 sub-clouds of 32 points, each using Farthest Point Sampling and K-Nearest Neighbors, and a discrete Variational Auto-Encoder (dVAE) is trained to encode these local patches into discrete tokens, as illustrated in Figure 5. Similar to Transformers (Vaswani et al., 2023) in natural language processing, the point cloud is represented as a sequence of patch embeddings, that can be processed by Transformer models. In addition, the special token [cls] – short for classification token, it is designed

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-331-2026 | © Author(s) 2026. CC BY 4.0 License.

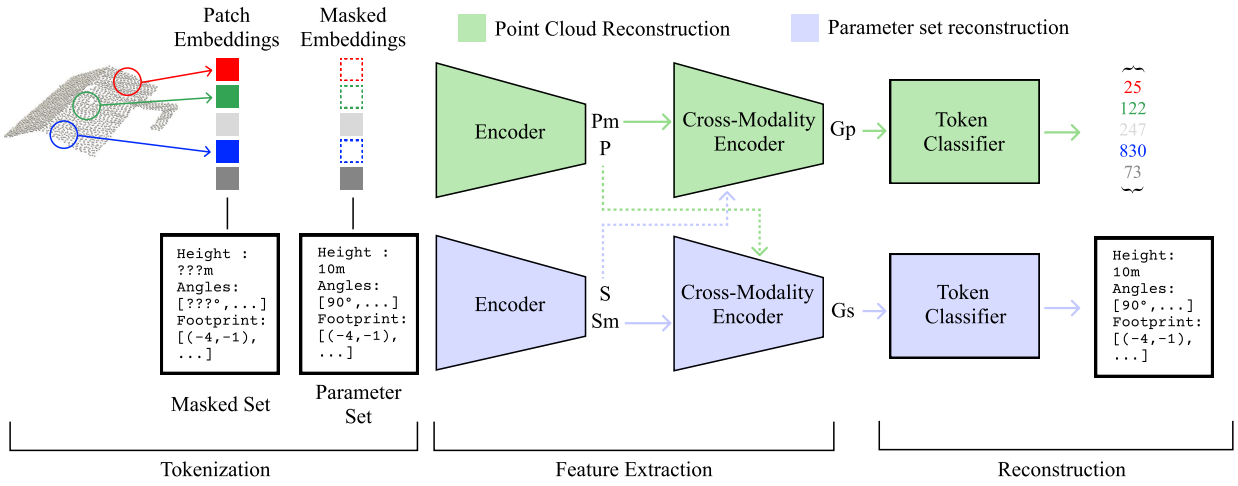


Figure 4. Point2WSS Architecture. *The green and purple lines show the reconstruction process of point cloud and parameter set respectively. The dotted lines indicate the cross-modal input of the unmasked modality.*

to capture semantic information from the whole point cloud – is added at the beginning of the token sequence, which is later used for multimodal alignment described in Section 4.4. This sequence can be partially masked, allowing a BERT-like model to learn to reconstruct the original sequence.

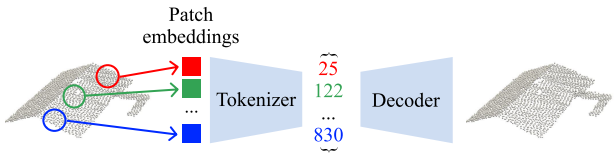


Figure 5. dVAE tokenizer architecture.

Masking. The point cloud masking strategy, following the approach introduced in Point-BERT (Yu et al., 2022), takes inspiration from the pretraining strategy used in BERT-based image transformers (Bao et al., 2021). The key idea is that randomly masking isolated point cloud patches may fail to conceal semantically meaningful information, since adjacent patches often contain redundant geometric features. To address this issue, a spatially coherent masking scheme is adopted, where neighbors patches, forming a continuous local region, are masked together. This encourages the model to learn both local geometric consistency and global structural relationships within the point cloud. To perform masking, patch tokens in the sequence are replaced with a designated [mask] token.

Reconstruction. The input point cloud is first divided into 64 sub-clouds of 32 points each, following the dVAE architecture. Each sub-cloud is processed by a mini-PointNet to generate an embedding sequence $w = \{w_{cls}, w_1, ..., w_{64}\}$. The resulting masked sequence of patch embeddings is passed through the point cloud encoder to produce masked point cloud features P_m. In parallel, the parameter token sequence is processed by the parameter set encoder to obtain $S = \{s_{start}, s_1, ..., s_n, s_{end}\}$. The pair (P_m, S) is then fed into the point cloud cross-modal encoder to generate G_p, where S serves as the source to compute cross-attention. G_p is passed through a token classifier composed of a fully connected layer followed by a softmax activation to obtain discrete tokens prediction y_{rec}. The ground-truth tokens are obtained by passing w trough the dVAE tokenizer. Since the tokens are discrete, a standard cross-entropy loss, named $\mathcal{L}_{\text{p}}$, is used for the point cloud reconstruction branch, only on masked tokens.

$$\mathcal{L}_p = \text{CE}(y_{rec}, y) \tag{1}$$

where $y_{rec} \in \text{R}^{B \times S}$: predicted tokens
$y \in \text{R}^{B \times S}$: ground-truth tokens
$\text{CE}(\cdot)$: cross-entropy
B: batch size, S: point cloud sequence length

4.3 Parameter Set Reconstruction

This branch is designed to predict a variable number of continuous parameters via a cross-attention with a point cloud embedding. An efficient strategy is to convert the variable number of continuous parameters into a token sequence and mask tokens corresponding to parameter values, as illustrated in Figure 6. The predicted parameter values will be used as input of the WSS parametric building model, introduced in Section 3, allowing the generation of a building mesh from a point cloud. In practice, only height and angles are masked, as the footprint, generally used to extract the building point cloud, is assumed to be known or can be inferred.

Tokenization. The parameter token sequence is designed to be both compact and unambiguous. Angles and the footprint should be encoded in a way that allows the model to understand which edge corresponds to each angle. A simple and effective strategy is to represent the footprint as a sequence of points coordinates, with the angles corresponding to each edge interleaved within the sequence. Since all parameters are continuous, the parameter set is tokenized into categories : parameter-type tokens and numerical tokens. Three parameter-type tokens are introduced – [height], [angle] and [point] – and defined to specify the type of parameter represented by the subsequent numerical tokens. Initially introduced to predict continuous molecular properties using Transformers with masked language modeling (Born and Manica, 2022), a sequence of numerical tokens – $[i\_j] = i \times 10^j$ – enable to encode continuous value and to capture the relative scale of numerical values. The value of an encoded parameter is obtained by summing the numerical tokens that appear until the next parameter-type

token. Four supplementary special tokens, [x], [y], [-] and [+] are employed to encode x and y point coordinates with numerical tokens. Moreover, classical tokens in natural language processing, [start] and [end] are added at the start and at the end of the token sequence. Similar to the point cloud [cls] token described in Section 4.2, the [start] token is used to capture the overall semantics of the parameter sequence, which is later used for multimodal alignment described in Section 4.4. The [end] token simply indicates the end of the sequence. To ensure a bijection between token sequences and parameter sets, the sequence follows a consistent structure: it starts with the height description and then alternates between point and angle descriptions, beginning from the uppermost-left point of the footprint, in the counterclockwise order. In this way, the token sequence is unambiguous, motivated by the assumption that such consistency would facilitate the model's ability to reconstruct masked token sequences. To reduce sequence-level bias, each parameter value is encoded using the same number of numerical tokens, ensuring a consistent representation. This prevents the model from inferring the magnitude of a continuous value based on token count rather than its actual encoded content. Furthermore, this design allows creating, during inference, a masked sequence from a given footprint without requiring any prior assumptions about the height or the angle set.

Masking. During training, a simple random masking strategy is used, as each token in the sequence carries meaningful information and is not redundant, by replacing tokens with a designated [mask] token. During validation and inference, only the numerical tokens corresponding to the height and angles are masked. This allows the model to predict a complete, variable-length set of continuous parameters.

Reconstruction. The parameter set reconstruction branch is symmetric to the point cloud reconstruction branch. Regression Transformers (Born and Manica, 2022) have shown that a token classification approach can outperform standard regression models. Therefore, a single cross-entropy loss, named $\mathcal{L}_{\mathrm{s}}$, is used, only on masked tokens, for training this branch.

$$\mathcal{L}_s = \mathrm{CE}(y_{rec},\, y) \tag{2}$$

where $y_{rec} \in \mathrm{R}^{B\times S}$: predicted tokens
$y \in \mathrm{R}^{B\times S}$: ground-truth tokens
$\mathrm{CE}(\cdot)$: cross-entropy
B: batch size, S: parameter set sequence length

4.4 Multimodal Alignment

Following (Kwon et al., 2023), two additional tasks are adopted to facilitate multimodal alignment, complementing the point cloud and parameter set reconstruction tasks, at both modal and cross-modal levels. Indeed, aligning the model representations of point cloud and parameter set facilitate accurate reconstruction of both modalities.

Modal Alignment. Out on the point cloud and parameter set encoders, two fully connected layers are used to project the embeddings of the point cloud [cls] token and parameter set [start] token into a shared dimensional space, respectively z_p and z_s, to compare the distance between their latent representations. The modal loss is used to bring corresponding representations closer together and is computed as:

$$\mathcal{L}_m = \frac{1}{2}\Big(\mathrm{CE}(L, y) + \mathrm{CE}(L^\top, y)\Big),\ \text{with } L = s\, z_p\, z_s^\top \tag{3}$$

where $z_p \in \mathrm{R}^{B\times D}$: [cls] token embedding
$z_s \in \mathrm{R}^{B\times D}$ [start] token embedding
s: learnable scalar
$y \in \{0,1\}^B$: labels for matching pairs
$\mathrm{CE}(\cdot)$: cross-entropy
B: batch size, D: embedding dim

Cross-Modal Alignment. Out of the point cloud and parameter set cross-modal encoders, the point cloud [cls] token and parameter set [start] token embeddings are fused by computing the element-wise product z_f. A fully connected layer, followed by a softmax, is then used to predict whether the pair is aligned. Pairs are generated by combining each element of a batch with all others. A weighted cross-entropy loss is applied, as the number of negative pairs significantly exceeds the number of positive pairs.

$$\mathcal{L}_f = \mathrm{CE}(\mathrm{FC}(z_f),\, y;\, \mathrm{weight} = [0.1, 0.9]) \tag{4}$$

where $z_f \in \mathrm{R}^{B^2\times D}$ = fused features
$\mathrm{FC}(\cdot)$: fully-connected layer
$y \in \{0,1\}^{B^2}$: labels for matching pairs
$\mathrm{CE}(\cdot)$: cross-entropy
$\mathrm{weight} = [0.1, 0.9]$: class weights
B: batch size, D: feature dim

4.5 Training Objective

To ensure good balance between reconstruction losses $\mathcal{L}_p$, $\mathcal{L}_s$ and alignment losses $\mathcal{L}_m$, $\mathcal{L}_f$, the total loss $\mathcal{L}_t$ is computed as follow :

$$\mathcal{L}_t = \mathcal{L}_p + \mathcal{L}_s + \mathcal{L}_m + \mathcal{L}_f \tag{5}$$

5. Experiments

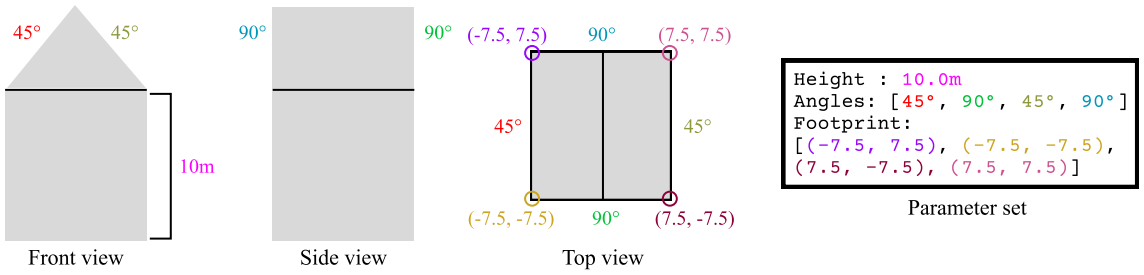
5.1 Synthetic Dataset

The training dataset should consist of pairs of point clouds and parameter sets. Generating these pairs directly from HD LiDAR point clouds would require a time-consuming labeling process, as training a multimodal model require a substantial amount of data. Instead, a pipeline for generating synthetic point cloud–parameter set pairs is proposed. It consists of three main steps: (i) building information retrieval, (ii) building generation, and (iii) LiDAR simulation.

Building Information Retrieval. To make the synthetic point cloud-parameter set pairs more realistic, information from the IGN building database[‡] is used, which contains building footprints and heights for the entire French territory. To ensure realistic angle configurations, an angle is assigned to each footprint edge, chosen from 0°, 90°, or an integer between 20° and 45°. Note that an angle of 0° in the configuration results in a flat roof, while an angle of 90° produces vertical roof segments.

Building Generation. This step is straightforward, since buildings are generated using the method presented in Section 3.

[‡] https://geoservices.ign.fr/bdtopo



[start][height][0_2][1_1][0_0][0_-1][point][x][-][0_2][0_1][7_0][5_-1][y][+][0_2][0_1][7_0][5_-1][angle][4_1][5_0][point][x][-][0_2][0_1][7_0][5_-1][y][-][0_2][0_1][7_0][5_-1][angle][9_1][0_0][point][x][+][0_2][0_1][7_0][5_-1][y][-][0_2][0_1][7_0][5_-1][angle][4_1][5_0][point][x][+][0_2][0_1][7_0][5_-1][y][+][0_2][0_1][7_0][5_-1][angle][4_1][5_0][end]

Parameter token sequence

[start][height][mask][mask][mask][mask][point][x][-][0_2][0_1][7_0][5_-1][y][+][0_2][0_1][7_0][5_-1][angle][mask][mask][point][x][-][0_2][0_1][7_0][5_-1][y][-][0_2][0_1][7_0][5_-1][angle][mask][mask][point][x][+][0_2][0_1][7_0][5_-1][y][-][0_2][0_1][7_0][5_-1][angle][mask][mask][point][x][+][0_2][0_1][7_0][5_-1][y][+][0_2][0_1][7_0][5_-1][angle][mask][mask][end]

Masked token sequence

Figure 6. Building’s parameter set tokenization.

LiDAR Simulation. To save computational time and introduce occlusion effects in the synthetic data, 1 km × 1 km scenes are generated using the first two steps. These scenes contain generated buildings and the corresponding ground, which is retrieved by triangulating a Digital Terrain Model[§] (DTM) produced by IGN. The DTM is a single-channel image providing elevation values at a one-meter resolution. A LiDAR simulator (Marchand et al., 2021), configured with realistic parameters, is used to simulate the LiDAR acquisition of the scenes. The simulated point clouds contain around 8 points per meter square. Building point clouds are extracted from scene point clouds by retaining points within a 1-meter buffer around each building footprint. This ensures that a few ground points are retained, which are expected to be useful for predicting the building’s height.

Simulated areas were chosen from dense urban environments of major French cities (e.g., Paris, Marseille, Bordeaux, Lyon) to increase the presence of occlusions, resulting in a challenging dataset. A total of 800,000 building point cloud–parameter set pairs were generated. The dataset was split into 80% for training, 10% for validation, and 10% for testing.

5.2 Training Setup

Computational Resources. Model training was performed on two NVIDIA GeForce RTX 4090 GPUs with distributed data parallelism to ensure efficient computation.

Preprocessing. Point clouds are either truncated or padded to 2048 points to enable uniform batch processing. The padded points are excluded from all computations. Moreover, point clouds are randomly rotated at each epoch to improve the model’s robustness to orientation variations and are centered around their centroid to avoid positional bias.

dVAE Training. The dVAE, used as point cloud tokenizer, is trained for 300 epochs of batch size 512, learning rate 1×10^{-3} and weight decay 5×10^{-4} with cosine warmup and cosine learning rate schedule. The dVAE is trained on synthetic data and, as it is a self-supervised model that does not require labeled data, an additional 500,000 building point clouds from HD LiDAR are extracted for training.

Point2WSS Training. The model is trained for 200 epochs of batch size 512, learning rate 1×10^{-3} and weight decay 5×10^{-4} with cosine warmup and cosine learning rate schedule, on the synthetic data.

5.3 Evaluation Setup

Evaluation Metrics. To evaluate the building reconstruction method, the following metrics will be used.

- Mean Absolute Error (MAE), to evaluate the error between predicted parameters and ground truth parameters;
- Chamfer Distance (CD) measuring the symmetric distance, in meter, between the input point cloud and a sampled point cloud from the reconstructed building mesh;
- Number of triangles (#T) to evaluate the mesh complexity.

Evaluation Datasets. The following datasets are considered in this work.

- Test subset of the synthetic dataset, introduced in Section 5.1, consisting of around 80,000 buildings;
- Around 350,000 building point clouds extracted from HD LiDAR and their corresponding footprint extracted from the IGN building database.

5.4 Results and Discussions.

Parameters prediction. First, the capacity of Point2WSS to predict a variable number of continuous parameters is evaluated on the synthetic test subset. The numerical tokens corresponding to the height, the footprint and the angle set are masked. The reconstructed tokens are then mapped back to their corresponding parameter value.

The results, presented in Table 1, show that Point2WSS is able to accurately predict a variable number of continuous parameters. The standard deviations are relatively high, due to the token

[§] https://geoservices.ign.fr/rgealti

Parameter	MAE
Height (m)	0.23±0.9
Angles (°)	2.34±5.35
Footprint (m)	0.41±0.93

Table 1. Parameter prediction accuracy.

classification approach. Indeed, the impact of a misclassified numerical token depend on its position. For instance, a misclassified token embedding the tens digit induces a greater error than one embedding the units digit, while the parameter set loss $\mathcal{L}_s$, introduced in Section 4.3, penalizes the model equally for both errors.

Building reconstruction. Second, the building reconstruction performance is evaluated in comparison with the state-of-the-art method Roofer[¶] (Peters et al., 2022), presented in Section 2.1. A masked parameter set sequence is generated from the footprint, as illustrated in Figure 6, and Point2WSS reconstructs the masked tokens corresponding to the height and angles. The WSS parametric building model is used to generate the building from the footprint, the predicted height and the predicted angles.

Dataset	Synthetic		HD LiDAR	
Method	Roofer	Ours	Roofer	Ours
μ_{CD}	**1.25**	1.43	**0.97**	2.14
$Q_{1,CD}$	0.63	**0.59**	**0.57**	0.98
$Q_{2,CD}$	**0.95**	**0.95**	**0.73**	1.55
$Q_{3,CD}$	**1.48**	1.72	**1.01**	2.45
σ_{CD}	**1.08**	1.72	**0.93**	2.23
$\mu_{\#T}$	24.9	**17.2**	25.6	**17.3**
$\sigma_{\#T}$	17.1	**4.8**	19.2	**5.2**

Table 2. Building reconstruction statistics.

The reconstruction results, presented in Table 2, show that the proposed reconstruction process achieves performance comparable to Roofer on synthetic data. Although Roofer performs slightly better due to its design optimized for point cloud fitting. Since Point2WSS predicts continuous value, small errors in these prediction could lead to larger CD, especially for height estimation as illustrated in Figure 7, on both synthetic and HD LiDAR datasets.

Roofer, as a data-driven method, performs better on HD LiDAR data due to the higher point density compared to the synthetic dataset. In contrast, the proposed method performs less effectively on HD LiDAR data as the WSS parametric building model cannot reconstruct all buildings with its defined parameters. Discontinuous building heights results in large CD even when the angles are correctly estimated, as illustrated in Figure 8.

Furthermore, the WSS parametric building model is unable to represent sub-structures such as garages, as illustrated in Figure 9, or dormers, as illustrated in Figure 10 and are not represented in the synthetic training dataset. These elements confused Point2WSS to infer the correct height and angles.

[¶] rooferpy: https://github.com/3DBAG/roofer

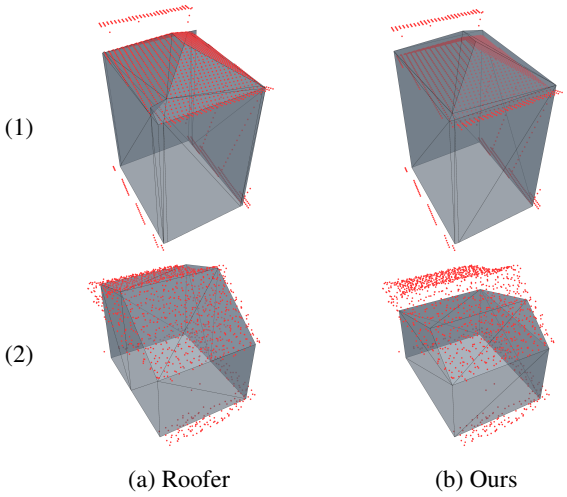


Figure 7. Height error. *(1) Synthetic / (2) HD LiDAR*

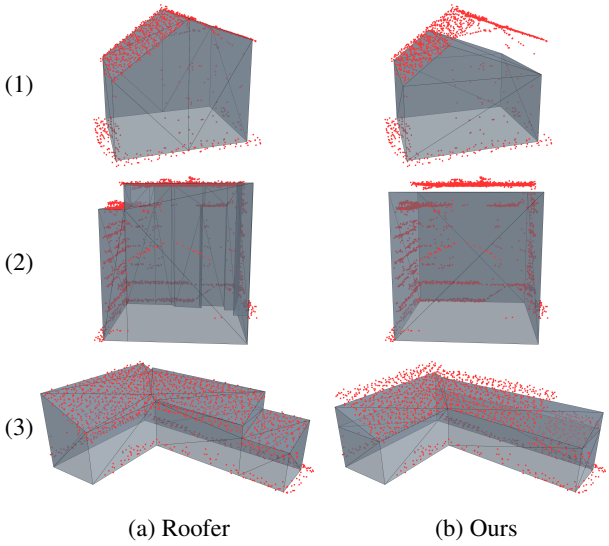


Figure 8. Buildings with discontinuous height. *HD LiDAR*

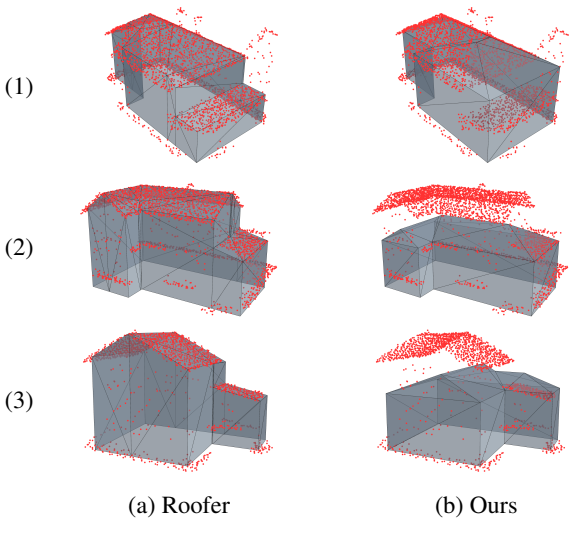
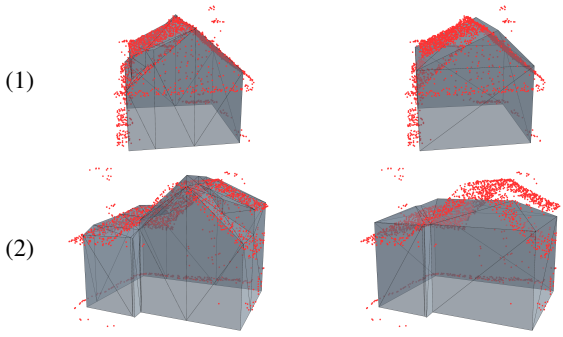


Figure 9. Buildings with garage. *HD LiDAR*

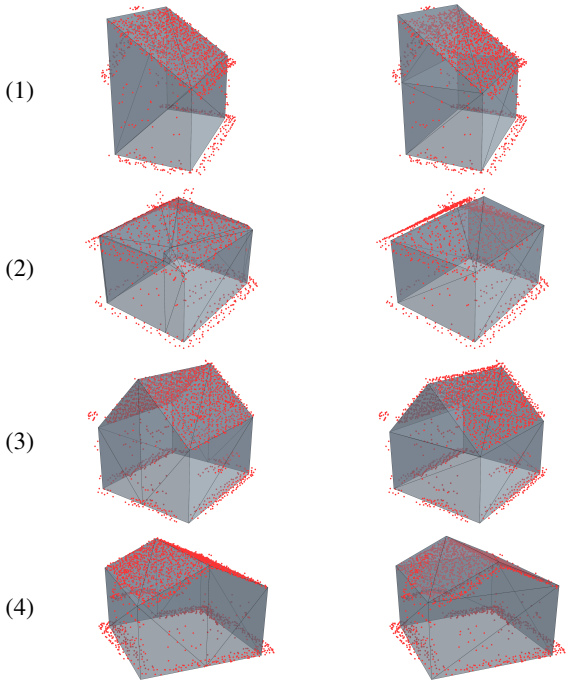
This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-331-2026 | © Author(s) 2026. CC BY 4.0 License.



(a) Roofer (b) Ours

Figure 10. Buildings with dormer. *HD LiDAR*

However, Figure 11 shows that buildings with hip or gable roofs are generally accurately reconstructed, validating that the training on synthetic data enables Point2WSS to perform reliably on HD LiDAR data.



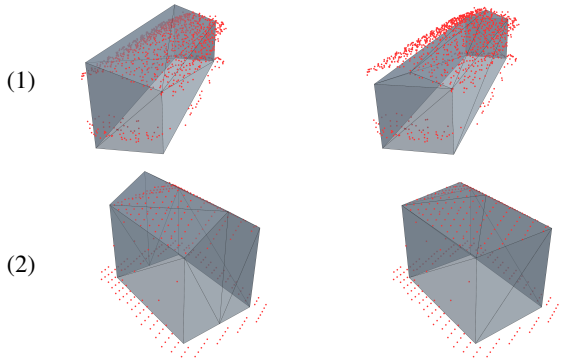
(a) Roofer (b) Ours

Figure 11. Buildings with shed, hip or gable roof. *HD LiDAR*

Roofer may fail to correctly reconstruct the complete planar configuration of the building, leading to missing planes, as illustrated in Figure 12, especially on point clouds with low density.

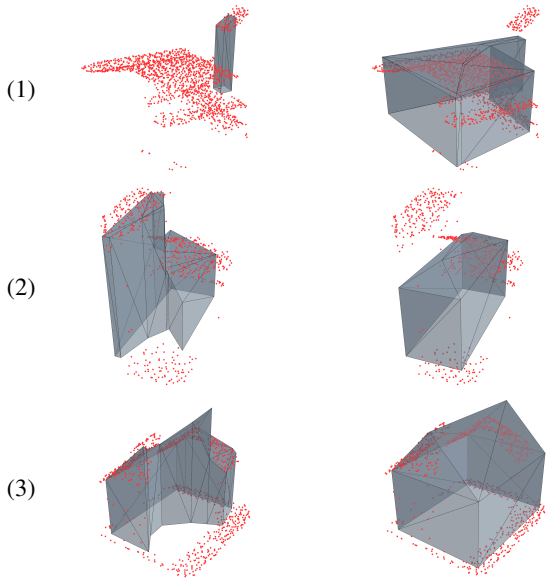
Finally, Roofer can generate unrealistic building structures, as illustrated in Figure 13, particularly when the point cloud is incomplete. In contrast, the proposed reconstruction method, lead to realistic building. The model is robust to incomplete data because, during training, it learned features that capture the essential geometry needed to infer angles and heights. By focusing on global shape patterns rather than relying on every detail, it can still produce realistic and coherent predictions from sparse or partial point clouds.

Mesh quality. Although it does not always perfectly fit the



(a) Roofer (b) Ours

Figure 12. Building reconstructed by Roofer containing missing planes. *(1) HD LiDAR / (2) Synthetic*



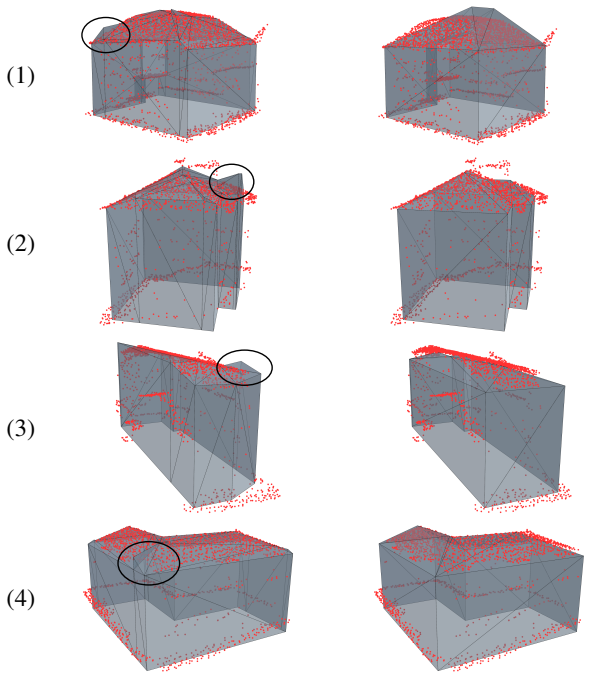
(a) Roofer (b) Ours

Figure 13. Buildings reconstructed by Roofer with unrealistic structures. *HD LiDAR*

input point cloud, the proposed method consistently generates realistic building structures with a lower polygonal density than Roofer, as presented in Table 2. Both methods rely on extrusion, assuming strictly vertical wall planes. However, Roofer can produce artifacts, as illustrated in Figure 14, by trying to overly fit the point cloud which degrade the overall mesh quality and hinder its use for electromagnetic simulation. In contrast, the WSS algorithmic approach produces clean structures.

Discussions. To summarize, while this parametric approach involves a slight trade-off in reconstruction fidelity, it ensures a structurally coherent and realistic building mesh. Furthermore, it achieves a lower triangle count compared to the meshes produced by Roofer, as demonstrated by the comparative metrics in Table 2. Crucially, unlike Roofer, which often introduces geometric artifacts as illustrated in Figure 14, our method produces clean, structured and manifold surfaces. This absence of artifacts, combined with a structural simplicity and lightweight mesh, makes our approach more suitable for radar simulation, where structural regularity and geometric simplicity are prioritized over raw point cloud fidelity.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-331-2026 | © Author(s) 2026. CC BY 4.0 License.



(a) Roofer (b) Ours

Figure 14. Buildings reconstructed by Roofer containing artifacts (circled in black). *HD LiDAR*

6. Conclusion and Perspectives

This paper aims to introduce a new paradigm for reconstructing, from point clouds, realistic LoD2.0 building meshes that are regularized, clean and have a low polygonal density through the WSS parametric building model. A multimodal model, Point2WSS, was trained on synthetic data to recover, from a given point cloud, a variable number of continuous building parameters, specifically the building height and the set of roof angles, which serve as input of the WSS parametric building model. The proposed method achieves performance comparable to the state-of-the-art method Roofer on synthetic data and scales reasonably well to HD LiDAR data, demonstrating that training Point2WSS on synthetic data can effectively generalize to real-world data. While Roofer achieves superior reconstruction performance on HD LiDAR data, it frequently produces geometric artifacts or unrealistic building structures, especially on sparse or incomplete point clouds, that can degrade the quality of electromagnetic simulations. In contrast, the proposed method involves a slight trade-off in reconstruction fidelity in favor of a structurally coherent, manifold and lightweight mesh, induced by the parametric building model. This makes it particularly relevant for radar applications where topological structure is prioritized over raw point cloud fidelity. This method was designed to be flexible and could be adapted to various building construction model, if input parameters could be represented as a token sequence.

Future works will focus on improving the synthetic dataset generation pipeline by better representing the different real-world building structure types and incorporating substructures like dormers, chimneys and garages, in order to make the synthetic dataset more representative of HD LiDAR data. Moreover, extending the WSS parametric model by introducing additional parameters will be considered and other building construction methods, especially procedural building generation methods, will be explored in order to reconstruct a wider variety of building structures and substructures, such as garages, chimneys and dormers. Finally, the Point2WSS training will be optimized by investigating pre-training strategies, point cloud augmentation techniques and improved features extraction methods. In particular, special attention will be given to the point cloud tokenization process, which is currently performed using a naive segmentation strategy.

References

Bao, H., Dong, L., Wei, F., 2021. BEiT: BERT Pre-Training of Image Transformers. *CoRR*, abs/2106.08254.

Bauchet, J.-P., Sulzer, R., Lafarge, F., Tarabalka, Y., 2024. Simplicity: Reconstructing buildings with simple regularized 3d models. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 7616–7626.

Biljecki, F., Ledoux, H., Stoter, J., 2016a. Generation of Multi-LoD 3D city model in CityGML with the procedural modelling engine Random3DCity. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W1, 51–59.

Biljecki, F., Ledoux, H., Stoter, J., 2016b. An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems*, 25–37.

Born, J., Manica, M., 2022. Regression Transformer: Concurrent Conditional Generation and Regression by Blending Numerical and Textual Tokens. *CoRR*, abs/2202.01338.

Boulch, A., Marlet, R., 2022. Poco: Point convolution for surface reconstruction. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6302–6314.

Chen, Y., He, T., Huang, D., Ye, W., Chen, S., Tang, J., Chen, X., Cai, Z., Yang, L., Yu, G. et al., 2024a. Meshanything: Artist-created mesh generation with autoregressive transformers. *arXiv preprint arXiv:2406.10163*.

Chen, Y., Wang, Y., Luo, Y., Wang, Z., Chen, Z., Zhu, J., Zhang, C., Lin, G., 2024b. Meshanything v2: Artist-created mesh generation with adjacent mesh tokenization.

Held, M., Palfrader, P., 2017. Straight skeletons with additive and multiplicative weights and their application to the algorithmic generation of roofs and terrains. *Computer-Aided Design*, 92, 33-41.

Huang, J., Stoter, J., Peters, R., Nan, L., 2022. City3D: Large-scale building reconstruction from airborne LiDAR point clouds. *Remote Sensing*, 14(9), 2254.

Kada, M., McKinley, L., 2009. 3D building reconstruction from LiDAR based on a cell decomposition approach. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 3), W4.

Kutzias, D., von Mammen, S., 2023. Recent advances in procedural generation of buildings: From diversity to integration. *IEEE Transactions on Games*, 16(1), 16–35.

Kwon, G., Cai, Z., Ravichandran, A., Bas, E., Bhotika, R., Soatto, S., 2023. Masked vision and language modeling for multi-modal representation learning.

Lafarge, F., Mallet, C., 2012. Creating large-scale city models from 3D-point clouds: a robust approach with hybrid representation. *International journal of computer vision*, 99(1), 69–85.

Li, L., Song, N., Sun, F., Liu, X., Wang, R., Yao, J., Cao, S., 2022. Point2Roof: End-to-end 3D building roof modeling from airborne LiDAR point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 193, 17-28.

Li, M., Nan, L., 2021. Feature-preserving 3D mesh simplification for urban buildings. *ISPRS Journal of Photogrammetry and Remote Sensing*, 173, 135–150.

Li, P., Guo, J., Zhang, X., Yan, D.-M., 2023. Secad-net: Self-supervised cad reconstruction by learning sketch-extrude operations. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16816–16826.

Liu, Y., D’Aronco, S., Schindler, K., Wegner, J. D., 2021. Pc2wf: 3d wireframe reconstruction from raw point clouds.

Liu, Y., Obukhov, A., Wegner, J. D., Schindler, K., 2024. Point2Building: Reconstructing buildings from airborne LiDAR point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 215, 351-368.

Ma, W., Chen, S., Lou, Y., Li, X., Zhou, X., 2024. Draw step by step: Reconstructing cad construction sequences from point clouds via multimodal diffusion. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 27154–27163.

Marchand, Y., Vallet, B., Caraffa, L., 2021. Evaluating Surface Mesh Reconstruction of Open Scenes. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B2-2021, 369–376.

Müller, P., Wonka, P., Haegler, S., Ulmer, A., Van Gool, L., 2006. Procedural modeling of buildings. *ACM SIGGRAPH 2006 Papers*, 614–623.

Nash, C., Ganin, Y., Eslami, S. A., Battaglia, P., 2020. Polygen: An autoregressive generative model of 3d meshes. *International conference on machine learning*, PMLR, 7220–7229.

Nys, G.-A., Poux, F., Billen, R., 2020. CityJSON building generation from airborne LiDAR 3D point clouds. *ISPRS International Journal of Geo-Information*, 9(9), 521.

Overby, J., Bodum, L., Kjems, E., Iisoe, P., 2004. Automatic 3D building reconstruction from airborne laser scanning and cadastral data using Hough transform. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(01).

Parish, Y. I., Müller, P., 2001. Procedural modeling of cities. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 301–308.

Peters, R., Dukai, B., Vitalis, S., van Liempt, J., Stoter, J., 2022. Automated 3D Reconstruction of LoD2 and LoD1 Models for All 10 Million Buildings of the Netherlands. *Photogrammetric Engineering & Remote Sensing*, 88(3), 165-170.

Salinas, D., Lafarge, F., Alliez, P., 2015. Structure-aware mesh decimation. *Computer Graphics Forum*, 34number 6, Wiley Online Library, 211–227.

Schnabel, R., Wahl, R., Klein, R., 2007. Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, 26(2), 214-226.

Tang, J., Li, Z., Hao, Z., Liu, X., Zeng, G., Liu, M.-Y., Zhang, Q., 2024. Edgerunner: Auto-regressive auto-encoder for artistic mesh generation.

Trouvé, N., Cochin, C., Houssay, J., Everaere, E., Husson, X., Unger, K., Jouadé, A., Fabbri, R., Houret, T., Talibart, B., Lévêque, O., Dupuis, X., 2024. Emprise : Synthetic environment for sensor design and virtual qualification. *2024 International Radar Conference (RADAR)*, 1–5.

Uy, M. A., Chang, Y.-Y., Sung, M., Goel, P., Lambourne, J. G., Birdal, T., Guibas, L. J., 2022. Point2cyl: Reverse engineering 3d objects from point clouds to extrusion cylinders. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11850–11860.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I., 2023. Attention is all you need.

Wang, R., Peethambaran, J., Chen, D., 2018. Lidar point clouds to 3-D urban models : A review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(2), 606–627.

Wu, R., Xiao, C., Zheng, C., 2021. Deepcad: A deep generative network for computer-aided design models. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 6772–6782.

Xu, Y., Stilla, U., 2021. Toward building and civil infrastructure reconstruction from point clouds: A review on data and key techniques. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 2857–2885.

Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., Lu, J., 2022. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19291–19300.

Zeng, H., Wu, J., Furukawa, Y., 2018. Neural procedural reconstruction for residential buildings. V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (eds), *Computer Vision – ECCV 2018*, Springer International Publishing, Cham, 759–775.

Župan, R., Vinković, A., Nikçi, R., Pinjatela, B., 2023. Automatic 3D Building Model Generation from Airborne LiDAR Data and OpenStreetMap Using Procedural Modeling. *Information*, 14(7).