RooFormer: Reconstructing Detailed 3D Roof Models from High-resolution Remote Sensing Imagery using Transformer

Dayu Yu[a,b], Fan Ye[c], Peng Yue[b,d,e,f,*], Min Chen[a,g], Filip Biljecki[h,i,*]

[a]School of Geography, Nanjing Normal University, Nanjing, 210046, Jiangsu, China
[b]School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, 430079, Hubei, China
[c] School of Computer Science, China University of Geosciences , Wuhan, 430079, Hubei, China
[d]Collaborative Innovation Center of Geospatial Technology, Wuhan, 430079, Hubei, China
[e]Hubei LuoJia Laboratory, Wuhan, 430079, Hubei, China
[f]Hubei Province Engineering Center for Intelligent Geoprocessing (HPECIG), Wuhan University, Wuhan, 430079, Hubei, China
[g]Key Laboratory of Virtual Geographic Environment (Ministry of Education of PRC), Nanjing Normal University, Nanjing, 210046, Jiangsu, China
[h]Department of Architecture, National University of Singapore, Singapore, 117566, Singapore
[i]Department of Real Estate, National University of Singapore, Singapore, 119077, Singapore

Abstract

Building roofs are essential for various geographical analyses such as solar potential analysis and urban microclimate simulation. Despite growing demand, reconstructing detailed 3D roofs remains challenging due to the complexity of roof geometries and variations in architectural styles. This paper introduces RooFormer, an end-to-end learning framework for reconstructing detailed and textured 3D roof models in mesh format from high-resolution imagery. RooFormer consists of a MaskFormer branch, which identifies and focuses on roof features, and a MeshFormer branch, which predicts detailed roof meshes. In the MeshFormer branch, a local self-attention mechanism is employed to understand mesh features, and a positional embedding layer is designed to integrate geometric and texture features. In addition, to measure the geometric similarity between predicted meshes and ground truth, we develop a loss function that integrates terms from both image and mesh spaces. Compared to existing 3D metrics, the proposed geometric loss term more accurately reflects the geometric differences in meshes. Experiments show that its normalized height error of 0.014 is lower than the 0.034 error of state-of-the-art methods. Visually, the reconstruction accurately reflects the geometric contours and structures of roofs, even with slight occlusions. We also demonstrate its generalization by testing it across various areas. The framework promises to enable richer building modeling and analysis for a wide range of digital city applications.

Keywords: Roof reconstruction, Mesh prediction, Building modeling, Transformer, High-resolution imagery

1. Introduction

3D building models are widely used elements in digital cities, offering significant analytical value (Lehtola et al., 2022). They are required for many applications in disaster management and environmental analysis. Within high levels of detail (LOD) models, roofs serve as a complex and vital geometric component, providing important information for solar potential analysis, urban microclimate simulation, and energy efficiency assessments (Zhang et al., 2020; Lei et al., 2023; Yu et al., 2024). Despite increasing demand, reconstructing detailed 3D roofs remains a challenging task due to the intricate nature of roof geometries, variations in architectural styles, and the need for high-resolution data (Dehbi et al., 2021).

The extraction of 2D roof boundaries (i.e., building footprints) has long been an active topic in photogrammetry and remote sensing (RS) (Guo et al., 2021). Recent advances in deep learning have facilitated the development of extensive deep neural networks (DNNs) for extracting pixel-wise footprints from RS images, significantly enhancing automation (Diakogiannis et al., 2020; Li et al., 2021b). Besides pixel-wise extraction, vectorized roof boundaries and structure lines based on DNNs have also been explored (Zhao et al., 2022). Multiple datasets of building footprints on national and global scales have been derived from satellite images using DNNs, such as GlobalMLBuildingFootprints (Microsoft, 2024) and CBF (Huang et al., 2024). However, little attention has been paid to the reconstruction of detailed 3D roof models using DNNs from RS images.

Various types of RS data, including images (Vallet et al., 2011; Li et al., 2021a), point clouds (Jiang et al., 2023), and photorealistic meshes (Yu et al., 2023), have been utilized to extract 3D geometric information of building models. Among these, monocular depth estimation uses single-view images to infer building heights based on depth cues such as texture gradients, shadows, and defocus. However, the extracted heights must be combined with footprints to generate low-resolution 3D building models lacking roof structures. Points representing building geometry can be extracted from point clouds to derive roof structures via template matching or parametric modeling, but this method requires extensive prior knowledge and is limited to simple roof shapes (Vallet et al., 2011). Photorealistic meshes offer detailed 3D building models with precise tex-

*Correspondence: pyue@whu.edu.cn, filip@nus.edu.sg

tures, but high acquisition, processing, and storage costs limit their frequency and availability of update in many regions (Yu et al., 2022). In contrast, RS imagery (e.g., satellite and aerial images) provides wide coverage, frequent updates, and public accessibility (Dutta and Das, 2023). Thus, integrating single-view RS images with DNNs has great potential for the efficient and accurate reconstruction of detailed 3D roof models. To our knowledge, no studies have yet explored this approach comprehensively.

This paper focuses on automatic 3D reconstruction of roof models from high-resolution RS images. We propose RooFormer, a deep learning method utilizing Transformer architecture to directly predict detailed and textured 3D roof meshes from single-view images in an end-to-end manner. Within RooFormer, we introduce a local self-attention mechanism for mesh feature identification and a positional embedding layer to capture and integrate geometric and texture features, forming the core components of the MeshFormer branch. Additionally, a MaskFormer branch is designed to identify and focus on roof texture from images. The loss function incorporates terms from both image and mesh domains. Our proposed geometric loss function for meshes, compared to existing 3D metrics like Chamfer distance, more accurately describes differences between output and ground truth (GT) meshes. We show that RooFormer is remarkably effective in 3D roof reconstruction, both in visual and quantitative evaluations. In summary, the contributions of this paper can be listed as follows:

- The first attempt to reconstruct detailed, textured 3D roof meshes from single-view RS images using DNNs in an end-to-end manner.
- The design of an expressive MeshFormer branch that includes local self-attention mechanisms for mesh feature extraction and positional embedding layers for integrating geometric and texture features.
- A loss function that incorporates terms from both mesh and image domains to measure the geometric similarity between predicted and GT meshes.
- Comprehensive evaluation through quantitative metrics, visual performance, and ablation studies.

2. Related work

2.1. Building geometry extraction using deep neural networks

In the last decade, DNNs have been extensively used to extract building geometry. Through training, these networks automatically learn the semantic features of buildings from the data without need for prior knowledge. Existing methods and algorithms can be classified into two main categories: image-based and point-based approaches.

Image-based DNNs rely on texture or spectral features to extract building footprints from satellite or aerial RS images (Nurkarim and Wijayanto, 2023; Liu et al., 2022; Vandita Srivastava and George, 2024). Most studies focus on the semantic segmentation of RS images, where building footprints are derived by classifying each pixel and identifying those belonging to buildings (Diakogiannis et al., 2020; Li et al., 2021b; Cheng et al., 2024). These studies are typically based on convolutional neural network (CNN) architectures, such as UNet (Ronneberger et al., 2015) and DeepLab-v3 (Chen et al., 2016). Recently, transformer-based models and large language models (LLMs) have also been employed, further enhancing efficiency and generalization capabilities (Li et al., 2024). Furthermore, to determine the distribution and number of buildings, many instance segmentation methods focus not only on identifying building pixels, but also on classifying individual buildings. These methods are often based on two-stage object detection models, such as Mask R-CNN (Chen et al., 2023) and Refine-Mask (Yang et al., 2023). To further improve extraction performance, many studies develop strategies that integrate DNNs with multiple data types.

The above segmentation methods typically produce pixel-wise results with curved and irregular boundaries, requiring post-processing to generate regular polygonal shapes. To address this, many polygonal segmentation methods were designed to produce building footprints in a desirable vector format. For instance, Polygon-RNN introduced an LSTM-based architecture to predict the vector of vertex location (Castrejón et al., 2017). PolyMapper extended the capabilities of Polygon-RNN by generating multiple vectorized building footprints (Li et al., 2018). Li (2023) proposed a joint semantic-geometric learning method for extracting polygonal buildings from remote sensing images. Furthermore, to extract the inner structures of building roofs in an end-to-end trainable manner, a fast and parsimonious parsing method was proposed to generate vectorized planar roof structures from high-resolution RS imagery (Zhao et al., 2022).

Point-based DNNs extract 3D points from point clouds to capture the geometric structure of building facades and roofs. Jiang et al. (2023) utilized a graph neural network (GNN) to extract 3D structural points of buildings from point clouds. Additionally, many DNNs have been designed for the semantic segmentation of point clouds, predicting the category of each point. These methods can be classified chronologically into four categories: projection-based networks (Su et al., 2015; Tatarchenko et al., 2018), voxel-based networks (Riegler et al., 2017), point-based networks (Charles et al., 2017), and Transformer-based networks (Zhao et al., 2021). Typically, Transformer-based networks achieve higher accuracy but come with increased computational complexity.

In summary, relative to image-based methods that extract building footprints, point cloud-based methods can capture the 3D geometric structure of buildings, providing more detailed information about their shape and structure. However, the collection and processing of point cloud data tend to be more time-consuming and costly. These methods, whether based on images or point clouds, can only extract basic building geometry such as footprints, facades, and roof points. However, additional complex steps, such as contour extraction and surface reconstruction, are still required to generate complete 3D building models (Kölle et al., 2021).

2.2. 3D Building reconstruction from RS data

In the context of 3D building reconstruction, height is a crucial yet often unavailable data element. To address this, numerous monocular depth estimation methods have been proposed. These methods estimate digital elevation models (DEMs) from single-view RS images by leveraging depth cues such as texture gradients, shadows, and defocus. For instance, Amirkolaee et al. (2019) used a CNN architecture to estimate height values from aerial images. Madhuanand et al. (2021) utilized a self-supervised learning approach to jointly learn depth and pose information. Furthermore, the MTBR-Ne network (Li et al., 2021a) was designed to learn geometric properties and the relationships between key components of 3D building models during monocular depth estimation. This allows for the direct extraction of building footprints and roof heights from single-view aerial images. More recently, the Building3D network (Mao et al., 2023) was developed to generate LOD1 3D building models by predicting elevation and building footprints from monocular images.

For 3D building model reconstruction based on point clouds, parametric shape modeling is primarily used. Most buildings have relatively simple base outlines and roofs, which can be easily identified from low-density data when roof details (such as chimneys) are not considered. These simple shapes can then be used to generate 3D building models through parametric representation (Kada and Mckinley, 2009). For example, the bottom outline decomposition method was proposed for modeling buildings with distinct roof height discontinuities (Vallet et al., 2011). This method involves creating a slightly larger surface than the building's base, decomposing it into cells, discarding those with low overlap, and template-matching the remaining cells' point clouds with a preset roof shape library to generate the roof. Park et al. (2019) first classify LiDAR data to extract points reflecting roof surfaces and use these points to estimate building heights, thereby generating LOD1 3D building models. Generally, these methods show lower automation in inner-city and residential areas with complex roof structures and rely on auxiliary data such as cadastral surveys.

Additionally, some studies focus on the rapid construction of low-resolution 3D building models at LOD0 and LOD1 based on 2D vectors (Hongchao Fan and Neis, 2014; Lei et al., 2024). For example, Goetz et al. (2012) proposed a framework for automatically creating CityGML models from OpenStreetMap polygons. Some polygons include height attributes, allowing the building footprints to be extruded to generate LOD1 building models with flat roofs directly (Agugiaro, 2016). Based on this strategy, the VGI3D system was developed to crowdsource the generation of 3D building models from OpenStreetMap data, featuring functionalities such as import, reconstruction, visualization, and modification (Zhang et al., 2021). However, most footprints in OpenStreetMap do not include height attributes. To address this, Biljecki et al. (2017) used machine learning methods to predict building heights from footprints and semantic information, resulting in LOD1 building models with a mean absolute error of less than 1 meter. For generating higher LOD models, IndoorOSM was proposed to generate LOD3 3D building models with windows and indoor layouts from OpenStreetMap data, despite involving many manual works (Goetz, 2013).

In summary, current studies have not yet attempted to directly reconstruct detailed 3D roofs from single-view images. Existing methods that use monocular images and vector data can achieve large-scale, low-resolution representations for urban areas, but their accuracy and level of detail remain limited. Point cloud-based methods show potential for reconstructing 3D building models with fine geometric structures, but they face challenges such as complex processing workflows and difficulties with texture mapping when applied to the reconstruction of detailed 3D building models.

2.3. Transformer neural network

Transformer, which utilizes a self-attention mechanism to capture global information, has revolutionized natural language processing and large language models (LLMs) (Vaswani et al., 2017). Recently, transformer has also made strides in image and point cloud understanding. The ViT splits input images into 16x16 patches, treating each as a token to encode features for image classification (Dosovitskiy et al., 2021). Building on this, the Pyramid Vision Transformer introduces a hierarchical structure and memory-optimized spatial attention to generate a feature pyramid for semantic segmentation (Wang et al., 2021). Furthermore, the Swin-Transformer employs window-based attention in successive Transformer blocks, enabling multi-scale feature extraction and establishing itself as a versatile backbone in computer vision (Liu et al., 2021).

Since 3D point clouds with coordinates can be used for position embedding, the self-attention mechanism is particularly suitable for them. Point Transformer (Zhao et al., 2021) was thus designed for classification and dense prediction tasks on point clouds. This network applies self-attention locally, enabling scalability to large scenes with millions of points. Furthermore, the Stratified Transformer (Lai et al., 2022) additionally samples distant points as keys in a sparser manner, enlarging the effective receptive field and establishing direct long-range dependencies while incurring negligible extra computations.

In summary, Transformers have shown great promise in processing images and point clouds. However, vanilla Self-Attention in Transformers requires a high computational cost. To mitigate this, images are typically divided into fewer patches for attention calculation, while attention in point clouds is computed locally for k-nearest neighbors using k-d trees. In contrast, meshes, with their explicit adjacency relationships, are better suited for local self-attention computation. Therefore, this paper proposes MeshFormer, which leverages mesh topology to aggregate mesh features efficiently.

3. Methodology

3.1. RooFormer network

The RooFormer network is an end-to-end trainable framework for reconstructing a detailed 3D roof mesh from a single

high-resolution RS image. The overview of RooFormer is illustrated in Figure 1. RooFormer takes two main inputs: an RS image showing a roof and an initial 3D plane mesh. The image provides details about the shape and structure of the roof, while the initial 3D plane ensures a watertight and manifold topology, along with adjacency information.

RooFormer is a novel transformer-based 3D mesh predictor, comprising a *MaskFormer branch* and a *MeshFormer branch*. *MaskFormer branch* first encodes the RS image into perceptual features, which are utilized by a lightweight decoder to estimate an auxiliary roof mask. The concatenated roof mask and perceptual features are then cascaded to *MeshFormer branch* to infer the 3D roof mesh. *MeshFormer branch* contains multiple cascaded *MeshFormer modules* designed to enhance mesh details gradually.

Both the mesh predicted by each *MeshFormer module* and the roof mask are computed as auxiliary losses, weighted alongside the loss of the final predicted mesh. In the mesh loss function, we introduce a geometric similarity loss to mitigate inflated values resulting from vertex imbalances between GT and predicted meshes, particularly in cases with similar shapes.

3.2. MaskFormer branch

Unlike images in ShapeNet (Chang et al., 2015), which only contain the objects to be reconstructed, the RS images include various non-target pixels such as trees, cars, and impervious surfaces, alongside roof pixels. These background pixels may divert the network's attention away from the roof, making it ineffective in identifying key features and generalizability.

Thus, we design *MaskFormer branch*, which consists of a *Transformer encoder* and a *Lightweight decoder* for perceptual feature extraction and mask prediction, respectively. By back-propagating an auxiliary loss between the mask and the target mesh, the extracted perceptual features can focus on the roof region, significantly reducing the interference of background pixels.

Transformer encoder. As depicted in Figure 2, given an image with $N\times3$, we first adopt 4 SwinV2_S layers (Liu et al., 2021) to encode it into high-dimensional perceptual features $F = \left\{\frac{N}{4}\times c_1, \frac{N}{8}\times c_2, \frac{N}{16}\times c_3, \frac{N}{32}\times c_4\right\}$. Notably, other backbones, such as Bi-Former (Zhu et al., 2023), can also serve as the encoder. We observed that they demonstrate similar performance for this task.

Decoder consists only of MLP layers with the advantages of being lightweight and computationally efficient. Its formulation follows Equation 1:

$$\begin{aligned} F'_i &= MLP(c_i, c_h, F_i), \forall i \\ F'_i &= Interpolate\left(N/4, F'_i\right), \forall i, \end{aligned} \tag{1}$$

where $MLP(\cdot)$ refers to a linear layer with c_i and c_h as the input and output dimensions, respectively. $Interpolate(\cdot)$ refers to the spatial interpolation operation. Specifically, each layer F_i in the multi-layer perceptual features F encoded by the Transformer encoder is first expanded to a uniform dimension c_h by $MLP(\cdot)$, and then upsampled to restore the resolution to $N/4$ using $Interpolate(\cdot)$.

Mask Header is also designed to consist only of MLP parameters, which predict a roof mask based on features F'_i. The mask is concatenated into each layer of F, thereafter serving as an input to the MeshFormer branch. Its formulation follows Equation 2:

$$\begin{aligned} F' &= GELU(BN(MLP(4c_h, c_h, Cat(F'_i)))), \forall i \\ Mask &= Sigmoid(MLP(c_h, 1, F')) \\ F &= Cat(Cat(F_i, Interpolate(N/2^{(i+1)}, Mask))), \forall i, \end{aligned} \tag{2}$$

where $Cat(\cdot)$ and $BN(\cdot)$ refer to the concatenation and batch normalization operations, respectively. $GELU(\cdot)$ and $Sigmoid(\cdot)$ as activation functions.

3.3. MeshFormer branch

As shown in Figure 1, in *MeshFormer Module*, positional encoding is initially embedded into input multi-layer perceptual features by the *Positional Embedding Layer*. It then employs cascaded *Residual MeshFormer Blocks* to predict 3D roof models and refines them using a *Subdivision Layer*.

Suppose a 3D mesh is represented by a tuple $M = \{V, T\}$, where $V \in \mathbb{R}^{m\times3}$ refers to vertices, and $T \in V^{m_1\times3}$ geometric refers to primitives. m and m_1 denotes the number of vertices and geometric primitives, respectively. Based on T, the adjacency matrix $A_{ij} \in \mathbb{R}^{m\times m}$ and the set of edge $E \in \mathbb{R}^{m_2\times2}$ can be easily computed, where m_2 denotes the number of edges.

(1) *Positional Embedding*

To infer 3D roof models from RS images, it is necessary to establish the association between 3D models and perceptual features F. This is typically achieved by the intrinsic parameters of the camera used to capture RS images, which are often inaccessible. Therefore, the proposed *Positional Embedding* layer embeds vertices of model into perceptual features using UV mapping, as shown in Figure 3. Its formulation follows Equation 3:

$$\begin{aligned} F^v_i &= UVProject(V, F_i), \forall i \\ F_{fusion} &= Cat\left(Cat(F^v_i, V)\right), \forall i, \end{aligned} \tag{3}$$

where F_{fusion} refer to the vector that include both perceptual and geometric features, with a shape of $(|V|, c_1+c_2+c_3+c_4+3)$.

(2) *Residual MeshFormer Block*

The MeshFormer Block is composed of a *Multi-Head Mesh Self-Attention* mechanism and multi feed-forward networks (i.e., *MeshFormer Layer*). With thousands of primitives as inputs, directly applying vanilla global self-attention mechanism (Vaswani et al., 2017) incurs a computational cost of $O(m_1^2)$.

Multi-Head Mesh Self-Attention. To this end, we design a topology-based self-attention for 3D meshes. As shown in Figure 4, instead of attending to all primitives, each query only needs to consider its adjacent neighbors. It is an element-wise operation, where each vertex is treated as a token, and the multi-head attention mechanism acts on each vertex in V. In this way, the complexity is reduced to $O(\frac{m}{k}, k^2) = O(m\times k)$, where k is the average number of neighbors.

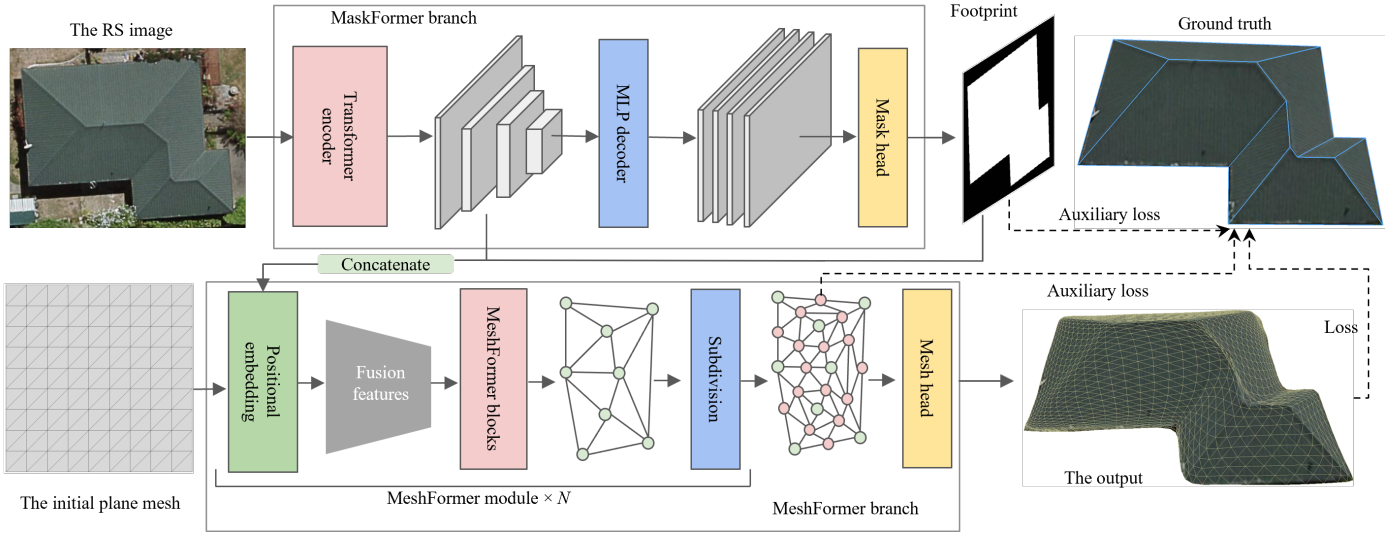


Figure 1: Overview of RooFormer network. The overall structure is a cascading architecture. The gray blocks are perceptual features of different stages, and the colorful blocks are different DNN modules.

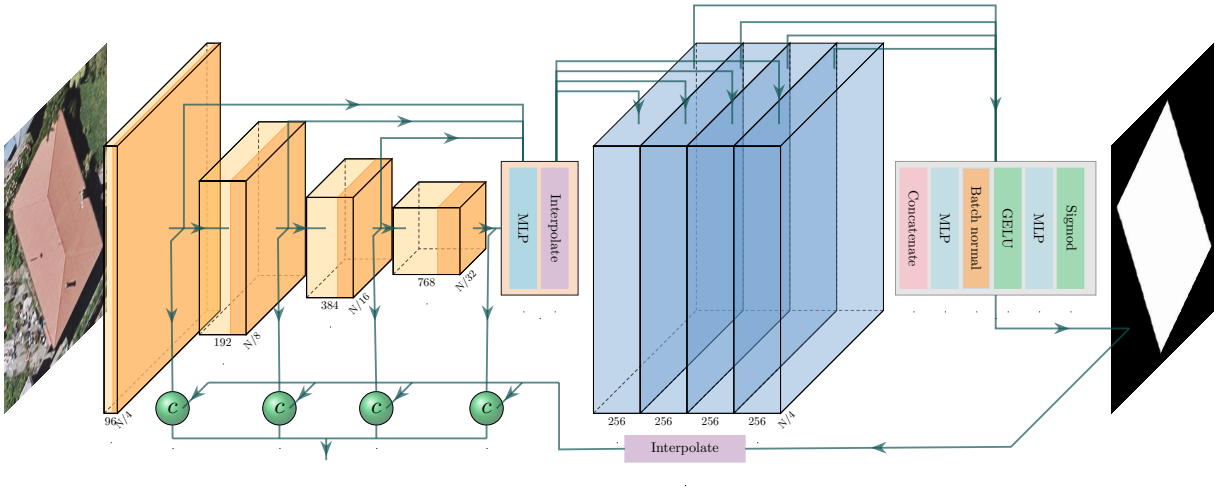


Figure 2: Overview of RooFormer network. The overall structure is a cascading architecture. The gray blocks in the figure are perceptual features of different stages, and the colorful blocks are different DNN modules.

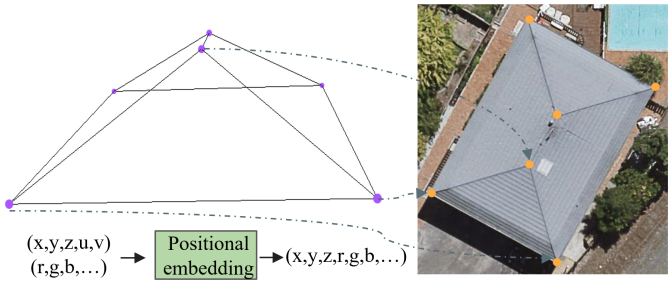


Figure 3: The illustration of positional embedding.

Formally, given that x_i is the feature vector of token v_i after positional embedding, the attention coefficient $c_{h,i}$ for the h-th head on v_i is formulated as per Equation 4:

$$
\begin{aligned}
&q_{(h,i)} = w_q x_i, \quad k_{(h,j)} = w_k x_j, \quad v_{(h,j)} = w_v x_j \\
&\alpha_{(h,i,j)} = q_{(h,i)} \odot k_{(h,j)} \\
&\hat{\alpha}_{(h,i,\ldots)} = softmax(\alpha_{(h,i,\ldots)}) \\
&c_{h,i} = \sum\nolimits_{j \in N(i)} \hat{\alpha}_{(h,i,j)} \times v_{(h,j)},
\end{aligned}
\tag{4}
$$

where w_q, w_v, and w_k are three trainable weights, $q_{h,i}$, $k_{h,j}$, and $v_{h,j}$ represent the Query, Key, and Value vectors respectively. $\odot$ denotes the vector dot product. $N(i)$ represents the set of neighboring indices of v_i, defined by the adjacency matrix A_{ij}, as shown in Equation 5.

$$
N(i) = \{j \,|\, A_{ij} = 1\}. \tag{5}
$$

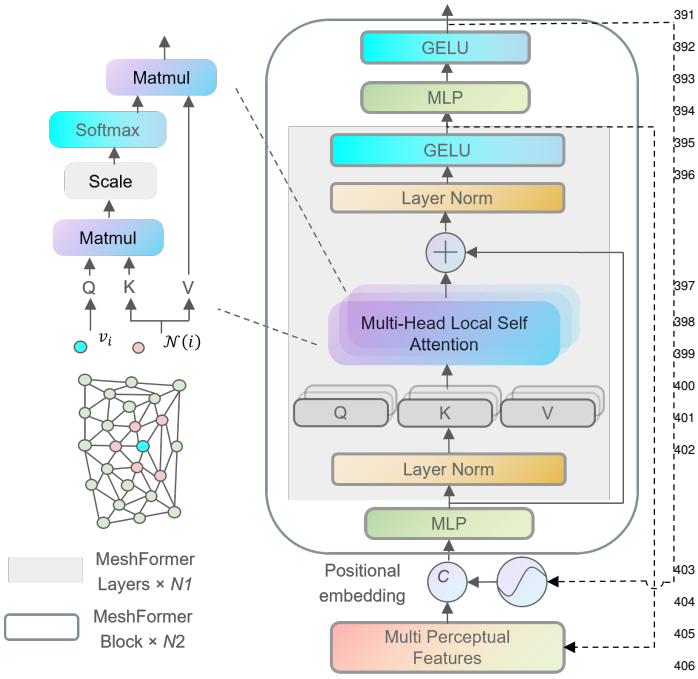


Figure 4: Structural illustration of *MeshFormer Layer* and *MeshFormer Block*.

MeshFormer Layer can be represented as per Equation 6. The feature vector x_i is first normalized by layer norm ρ. Then, the multi-head mesh self-attention coefficients are computed. As shown in the gray block in Figure 4, the layer integrates multi-head mesh self-attention and a residual connection, producing new feature vectors for all tokens as its output. The residual connection facilitates information exchange between feature vectors and improves gradient flow.

$$y_i = GELU(\rho(\frac{1}{H}\sum_{h=1}^{H} c_{h,i} + \rho(x_i))). \quad (6)$$

MeshFormer Layer is cascaded $N1$ times to form *MeshFormer Block*, shown in the box in Figure 4. To make *MeshFormer Block* also cascadeable, two MLP layers are added at the start and end to adjust the dimensions of inputs and outputs. The predicted mesh vertices from *MeshFormer Block* are embedded in the output of last MeshFormer layer to serve as input for the subsequent block.

(3) *Subdivision Layer and Mesh Head*

A key function of *Subdivision Layer* is to increase the details of meshes as required. To this end, the features F_f is processed by a MLP layer, then the features are mapped onto a higher-resolution 3D mesh via the half-edge subdivision. As shown in Equation 7, new vertices are added at the midpoint of each edge, resulting in the number of primitives changing from m_1 to $4 \times m_1$. Figure 5a illustrates the structure of *Subdivision Layer*.

$$\gamma_{up}(F_i, E) = 0.5(F_{\mathrm{f}}^i + F_{\mathrm{f}}^j), \forall (i,j) \in E. \quad (7)$$

Mesh Head is designed to predict the final 3D mesh at the specified resolution based on decimation, as illustrated in Figure 5b. The decimation is represented by Equation 8, where an edge subset E' uniformly sampled from E is collapsed at the half-edge, reducing the number of mesh vertices by $|E'|$ and the number of primitives by $2|E'|$. Meanwhile, we apply an *MeshFormer Layer* to map the feature F_f to the final mesh vertices. Both the subdivision and decimation operations can easily obtain manifold triangle and edge sets.

$$\gamma_{dn}(F_f, E') = F_f \setminus \{F_f^i, F_f^j\} \cup 0.5(F_f^i + F_f^j), \forall (i,j) \in E'. \quad (8)$$

3.4. Loss function

We formulate the training of RooFormer as a regression process, in which we minimize the surface difference between the predicted mesh and the ground truth. As defined in Equation 9, the loss function consists of the mesh loss l_{mesh} and the image loss l_{mask}.

$$L = \lambda_1 \sum_{i=1}^{N} l_{\mathrm{mesh}}(M_i) + \lambda_2 l_{\mathrm{mask}}, \quad (9)$$

where, λ_1 and λ_2 are pre-defined weights, M_i is the predicted mesh by *MeshFormer Block* or *Mesh Head*.

l_{mask} adopts cross-entropy loss to output precise building footprints and constrain the attention to roof pixels. l_{mesh} consists of the geometric similarity loss for the shape quality of meshes l_{geom} and the regularization loss l_{reg} for the topological quality of meshes, as per Equation 10.

$$l_{\mathrm{mesh}} = \lambda_3 l_{\mathrm{geom}} + l_{\mathrm{reg}}, \quad (10)$$

where λ_3 is a pre-defined weight.

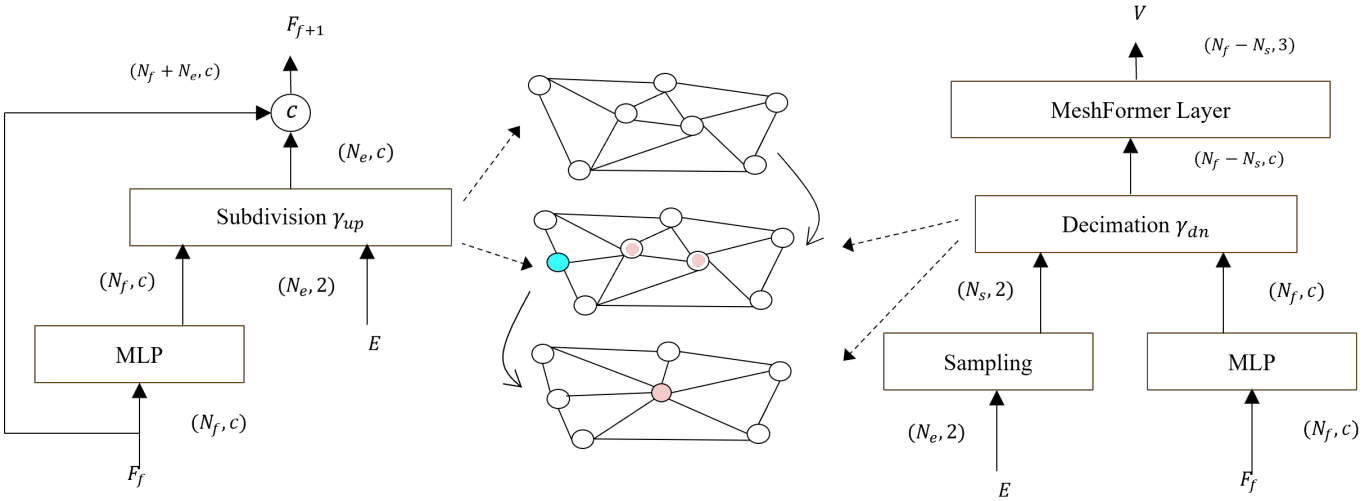
In CV for 3D, the Chamfer distance (CD) is commonly used to measure the similarity between two point sets (Yuan et al., 2021; Wang et al., 2018). As shown in Figure 6a, the predicted mesh and the GT mesh are very close in terms of geometric shape and spatial position. However, the number of vertices (gray points) in the predicted mesh is significantly greater than that in the GT mesh (blue points). In this case, the CD may be large even if the two sets are close. Due to the resolution change of 3D roof meshes after the *Subdivision Layer* and *Mesh Head*, it is hard to maintain an similar quantity to the GT vertices. To avoid the irregular fluctuations of L caused by CD, Our proposed l_{geom} measures surface similarity between 3D meshes rather than vertex similarity. It is defined as the average distance between two meshes in the Z-direction, as given in Equation 11.

$$\begin{aligned} V_1 &= \mathrm{sample}(M_1) \\ l_{\mathrm{geom}}(V_1, M_2) &= \frac{1}{|V_1|}\sum_{v \in V_1} \|v - I(v, M_2)\|_2, \end{aligned} \quad (11)$$

where M_1 and M_2 refer to the predicted 3D mesh and GT, respectively. $sample(\cdot)$ uniformly sample $N3$ points on the surface of a mesh. $I(\cdot)$ computes a unique intersection point where a ray is cast vertically from vertex v to M_2, which is a non-closed and space-continuous. In rare cases where there is no intersection between v and M_2, the position on M_2 closest to v is returned as the value, as shown in Figure 6b.

l_{reg} consists of Laplacian term l_{lap} and edge term l_{edge}, weighted as per Equation 12:

$$l_{\mathrm{reg}} = \lambda_4 l_{\mathrm{lap}} + \lambda_5 l_{\mathrm{edge}}, \quad (12)$$



(a) Subdivision Layer (b) Mesh Head

Figure 5: Structural illustration of (a) Subdivision Layer and (b) Mesh Head.

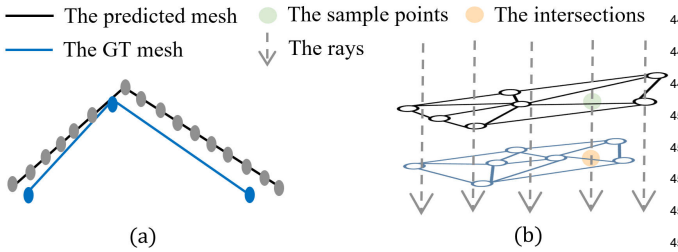


Figure 6: Illustration of (a) unbalanced vertices between the predicted and GT meshes and (b) the proposed l_{geom} loss.

where λ_4 and λ_5 represent the weights for l_{lap} and l_{edge} respectively.

l_{lap} is given by Equation 13. It penalizes vertices from moving too freely, ensuring that adjacent vertices have similar movement distances to avoid self-intersection of the mesh and excessive deformation during the training process.

$$l_{\text{lap}}(V) = \sum_{v \in V} \|\delta'_v - \delta_v\|_2^2$$
$$\delta_v = \sum_{k \in \mathcal{N}(v)} \frac{k}{\|\mathcal{N}(v)\|}, \tag{13}$$

where, $N(v)$ represents the set of vertices topologically adjacent to vertex v. l_{edge} is given by Equation 14 to prevent the generation of overly long edges.

$$l_{loc}(E) = \sum_{(i,j) \in E} \left\| V_i - V_j \right\|_2^2. \tag{14}$$

4. Experiments

4.1. Datasets and implementation details

We conducted experiments on the image-building paired (IMP) dataset (Ren et al., 2021) and various publicly available aerial images (introduced in more detail later) to verify the performance and robustness of the proposed method. The IMP dataset consists of 3,585 roofs paired with the input aerial image and 3D mesh. The 3D roof meshes, which vary in structure and complexity, were modeled manually in a semi-automated manner. Figure 7 shows some example roof meshes and aerial images. The 3D roof meshes retain only the shape-relevant edges/vertices by removing duplicates and redundancies. The number of vertices in all 3D roof meshes ranges from 5 to 34, and the image sizes range from 95×103 to 1066×848.

After removing 14 roof samples with topological errors, the dataset was randomly divided into 2,857 training samples and 714 testing samples. By applying horizontal and vertical flips to the training images and meshes, we obtained a total of 8,571 training samples. To facilitate batch processing during training, each image was resized to 224×224.

Our method was trained for 100 epochs with a batch size of 16 using the Adam optimizer on two Nvidia GeForce RTX 3090 devices with 48 GB of memory. The initial learning rate was set to 1×10^{-4}, and the final learning rate was 1×10^{-7}. We decayed the learning rate using MultiStepLR, with milestones at [40, 70, 90] and a gamma value of 0.5.

The *MeshFormer module* count N was 3, with $N1$ set to 1 *MeshFormer block* and $N2$ set to 6 *MeshFormer layers*. The number of self-attention heads was set to 4. Feature dimensions $c1, c2, c3$, and $c4$ were 96, 192, 384, and 768, respectively, with the input and output dimensions of the *MeshFormer layers* being 198. The loss weight coefficients $\lambda_1, \lambda_2, \lambda_3, \lambda_4$, and λ_5 were 1, 0.1, 1, 0.5, and 0.1, respectively. The number of sampled points in the l_{geom} loss function was set to 7000. The initial 3D mesh has 100 vertices, which increases to 361 after the first subdivision and to 1369 after the second subdivision.

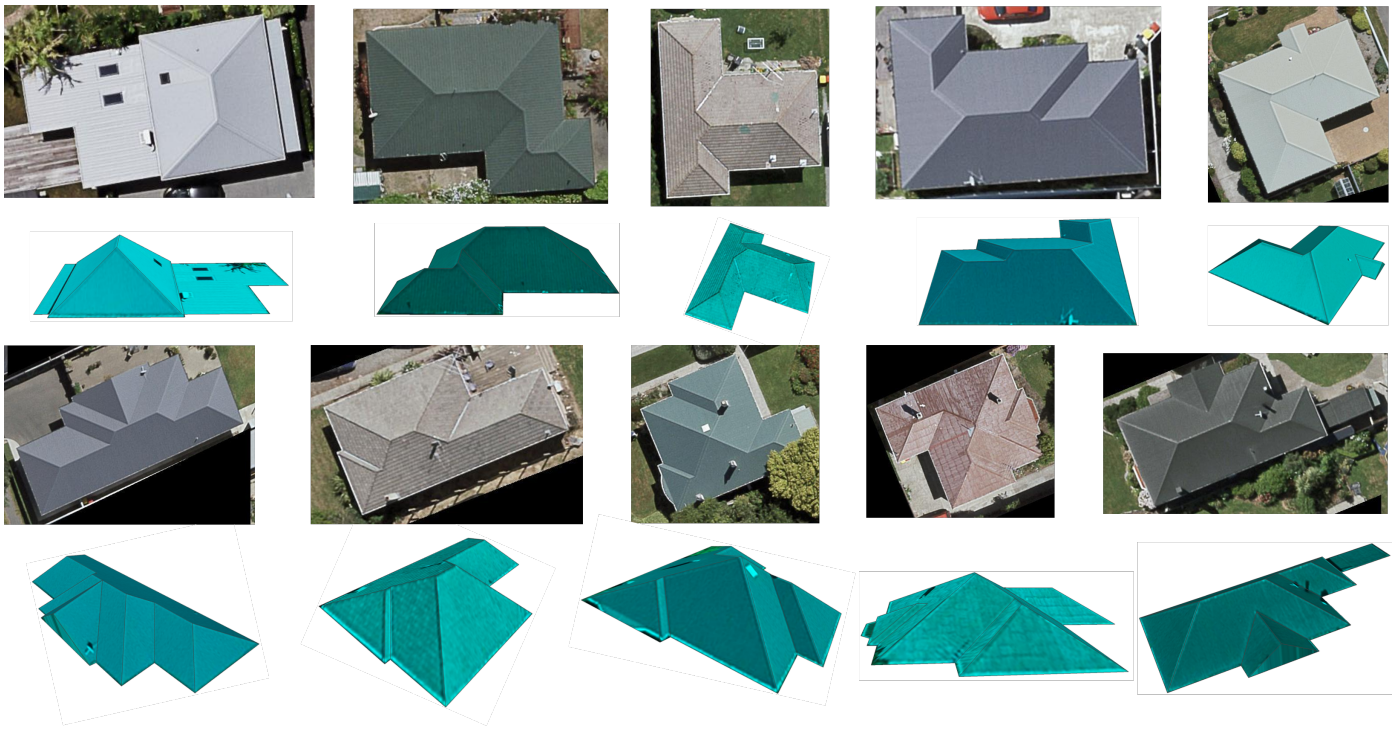


Figure 7: Example roof meshes paired with the corresponding aerial images (Ren et al., 2021).

4.2. Evaluation metrics

The output of 3D roof meshes from the considered methods were evaluated and compared using different metrics. We uniformly sampled points from the result meshes and GT meshes to calculate the Chamfer distance d_{cd}. Additionally, we calculated the percentage for surface distance threshold p_τ and $p_2\tau$. To better understand performance in terms of height, we report the mean absolute error z_{L1} and mean squared error z_{L2} in the z-axis. For the Chamfer distance, z_{L1}, and z_{L2}, smaller is better. For p_τ and $p_2\tau$, larger is better.

The Chamfer distance d_{cd} is defined as the sum of the distances from each point in a set to its nearest neighbor in another set, as per Equation 15.

$$d_{cd}(V_1, V_2) = \frac{1}{|V_1|}\sum_{x\in V_1}\min_{y\in V_2}\|x-y\|_2^2 + \frac{1}{|V_2|}\sum_{y\in V_2}\min_{x\in V_1}\|y-x\|_2^2, \tag{15}$$

where, V_1 and V_2 are the points uniformly sampled from the result mesh and its GT mesh, respectively. The normalized error z_{L1} and z_{L2} on the z-axis are defined as per Equation 16 and 17, respectively. Where, $\hat{z}$ and z_i are the height of the surface points from the result mesh and its GT mesh at the same positions, respectively.

$$z_{L1} = \frac{1}{m}\sum_{i=1}^{m}\frac{|\hat{z}_i - z_i|}{\max(z_i) - \min(z_i)}. \tag{16}$$

$$z_{L2} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\hat{z}_i - z_i}{\max(z_i) - \min(z_i)}\right)^2. \tag{17}$$

The percentage thresholds p_τ and $p_2\tau$ represent the percentage of points on the generated 3D roof mesh surface that are within a distance $\tau = 1e-4$ of the same positions on the GT model surface, as shown in Equation 18.

$$p_\tau(V_1, V_2) = \frac{1}{|V_1|}\sum_{x\in V_1}\sum_{y\in V_2}\delta\left(\|x-y\|_2^2\right) < \tau, \tag{18}$$

where, V_1 and V_2 are the surface points from the result mesh and its GT mesh at same positions, respectively. $\delta(\cdot)$ is an indicator function that is 1 when the distance is less than τ and 0 otherwise.

5. Results and discussion

5.1. Quantitative analysis

We compared our method with Pixel2Mesh (Wang et al., 2018), the state-of-the-art, open-source method available for 3D reconstruction from single images. The training and inference of Pixel2Mesh rely on the camera intrinsic parameters, which are not provided in the IMP dataset. To adapt Pixel2Mesh for IMP, we replaced its *Perceptual Feature Pooling Layer* with our proposed *Positional Embedding Layer*, creating a variant referred to as *P2M_PE*. Furthermore, for a fair comparison, we replaced the ellipsoid mesh used in Pixel2Mesh with a plane mesh, resulting in *P2M_Plane*. To validate the performance of the proposed geometric similarity loss l_{geom}, we replaced the Chamfer distance in Pixel2Mesh with l_{geom}, resulting in *P2M_Geom*.

Table 1 summarizes the results and comparisons on the IMP dataset. RooFormer achieved better performance on all metrics. *P2M_PE* achieved a $p_{2\tau}$ of only 18.97. Even after replacing the

Table 1: Comparison of different P2M variants

Method	d_{cd}	p_τ	$p_{2\tau}$	z_{L1}	z_{L2}
P2M_PE	0.12	9.55	18.97	0.069	0.592
P2M_Plane	1.61e-3	24.27	44.34	0.034	0.353
P2M_Geom	6.11e-4	46.80	72.25	0.023	0.207
RooFormer	3.57e-4	58.88	83.95	0.014	0.164

ellipsoid with a plane, $p_{2\tau}$ was only 44.34, which is significantly lower than the 83.38 achieved by RooFormer. Moreover, RooFormer achieved a relatively lower d_{CD} of 3.7×10^{-4} compared to other methods, showing the position accuracy of roof reconstruction. In terms of height errors, RooFormer's z_{L1} and z_{L2} are 0.014 and 0.164, respectively, about one-third of those of *P2M_Plane*. This indicates that the average elevation error of the reconstructed roofs with RooFormer is less than 0.05 meters when the roof height is 3 meters, whereas *P2M_Plane* has an average error exceeding 1 meter.

Compared to *P2M_Palne*, *P2M_Geom* achieved a $p_{2\tau}$ improvement of 27.91, confirming that the proposed l_{geom} loss is more suitable for 3D reconstruction tasks than point-wise distance. Furthermore, RooFormer improved $p_{2\tau}$ by 11.7 over *P2M_Geom*, demonstrating the superiority of the proposed network architecture and MeshFormer for this task.

In terms of training costs, RooFormer has a training efficiency similar to Pixel2Mesh but uses twice as much memory under the same conditions. However, if we replace the proposed multi-head mesh self-attention with vanilla self-attention, GPU memory usage increases 18 times. This significant increase is because it uses a topology-based local self-attention mechanism instead of a global one. This finding demonstrates the effectiveness of our designed multi-head mesh self-attention in inferring geometric features of 3D meshes.

5.2. Qualitative analysis

Figure 8 shows representative 3D roofs reconstructed by RooFormer and alternative methods on the validation set. RooFormer can reconstruct fine roof models of buildings without depending on preliminary boundary segmentation and any hand-crafted constraints. It properly reconstructs the geometric structures of complex roofs from single RS images. *P2M_Plane* performs poorly in terms of positional accuracy, boundary definition, and shape structure. *P2M_Geom* produces relatively finer shape structures, but some incorrect reconstructions around boundaries still exist. This may be caused by adjacent non-roof pixels such as impermeable surfaces and cars. In contrast, RooFormer reconstructs roof meshes with higher boundary accuracy because the *MaskFormer branch* predicts the roof mask to calculate an auxiliary loss.

In Figure 9, we display the normalized error distribution along the Z dimension for the results of RooFormer and P2M_Plane. Brighter colors indicate larger errors. Compared to P2M_Plane, the 3D roof meshes reconstructed by RooFormer exhibit smaller errors in the Z dimension. However, it is often observed that the errors at the ridge are larger than those in the planar areas.

To evaluate the structural integrity of the results, we computed and map the principal directions of curvature. The principal curvatures describe the amount of bending, while the principal directions describe the orientation of this bending. Typically, the structural lines of a roof lie in the areas with the highest curvature. In Figure 10, green indicates regions with low curvature and gentle slopes, while other colors represent areas with high curvature and significant bending. It is evident that the meshes reconstructed by RooFormer exhibit clear structural lines. Therefore, by clustering the principal curvatures, the reconstructed results can be directly applied to tasks such as the extraction of the roof structure line.

In addition to better geometric accuracy, RooFormer reconstructs roof meshes with superior topological quality. With the same number of primitives, the meshes reconstructed by *P2M_Plane* have an uneven primitive distribution, with an excessive concentrated around the roof's outline. This results in the sparse appearance of the roof wireframes in the last rows of Figure 8. In contrast, the wireframes of the roof mesh reconstructed by RooFormer are denser, reflecting a more uniform distribution of mesh primitives. Meanwhile, as shown in Figure 8, the input RS image can be directly mapped to the texture of the reconstructed mesh through the *Positional Embedding Layer*, without additional post-processing steps required by Pixel2Mesh.

Furthermore, we tested the proposed network on open aerial images obtained from the OpenAerialMap platform (https://openaerialmap.org, last accessed August 2024) for a qualitative evaluation of generalization. We used the model trained on the IMP dataset and directly inferred these aerial images without fine-tuning. As shown in Figure 11, the model trained on IMP data generalizes well to open aerial images from various countries.

5.3. Performance analysis

Table 2 presents the training and inference performance of RooFormer. On a single GPU, RooFormer can infer more than 225 images per second and generate the corresponding meshes. With two GPUs, a batch size of 8 enables training of more than 14 images per second, with 100 epochs taking nearly 4 hours and requiring a total of 9.43-17.68 GB of GPU memory. The input image size does not affect the inference efficiency, but significantly impacts the GPU memory usage during training. Furthermore, we evaluated the performance of the proposed Multi-Head Mesh Self-Attention. As shown in Figure 12, with an increase in the number of mesh vertices, the proposed Self-Attention outperforms Vanilla Self-Attention in both floating point operations per second (FLOPs) and GPU memory usage, with a more favorable growth rate.

5.4. Ablation studies

5.4.1. Ablation studies on at the network architecture level

We conduct controlled experiments to analyze the importance of key parameters and components in RooFormer, including the number of heads (MLH) and feature dimension (MLD) in the MeshFormer layer, the number of MeshFormer layers

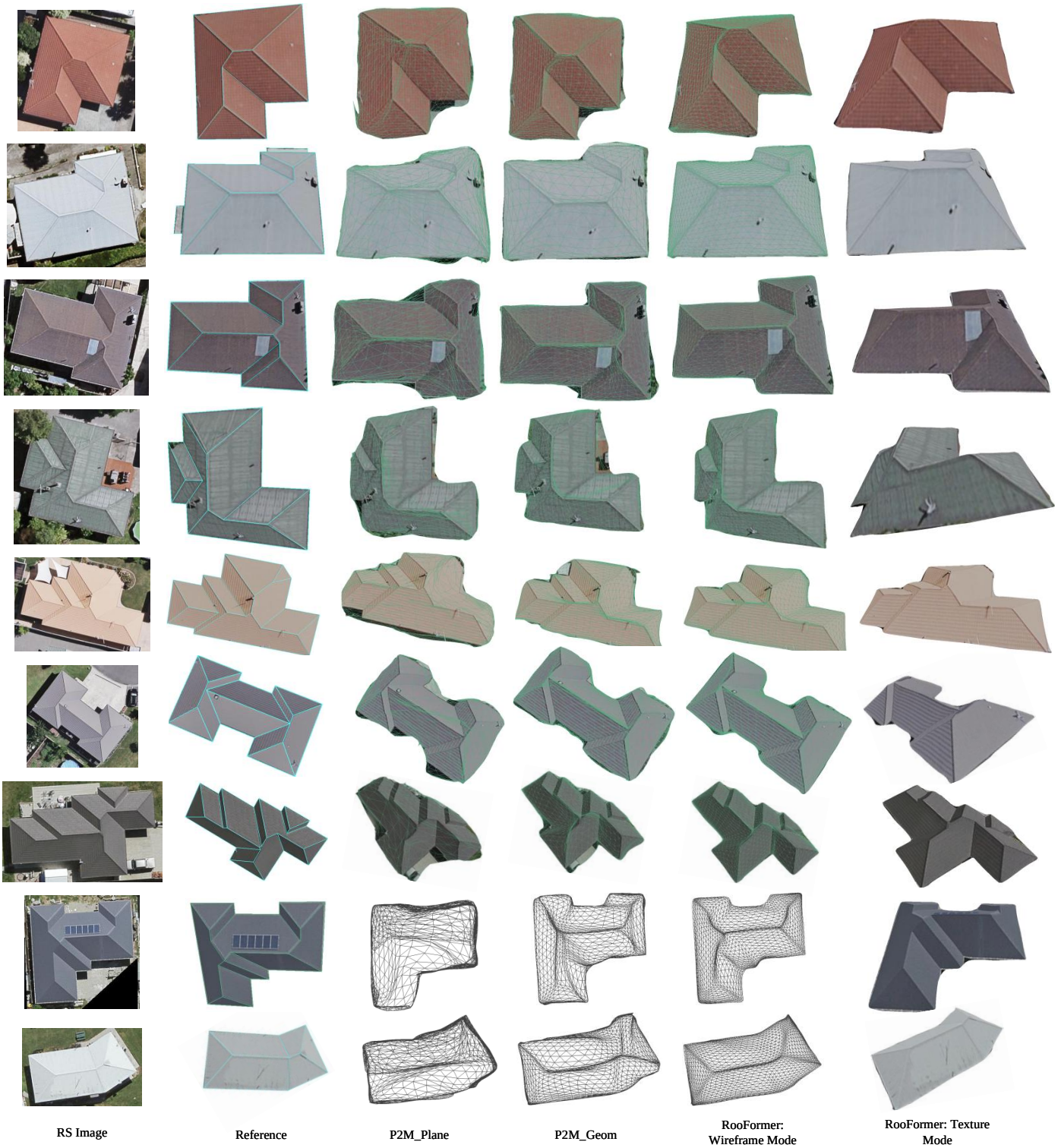


Figure 8: Comparative evaluations against competing methods on the validation set. From left to right: RS Image, Reference, *P2M_Plane*, *P2M_Geom*, and the wireframes and texture visualization modes of RooFormer.

(MLN), and the number of MeshFormer modules (MMN). Table 3 reports the performance under different settings used in the experiment.

As MLH, MLD, and MLN decrease, the parameter volumes and the evaluation metrics d_{cd}, p_{τ}, $p_{2\tau}$, z_{L1}, and z_{L2} decrease correspondingly. For example, when MLD is 118, the value of $p_{2\tau}$ is 81.94%, which is nearly a 2% decrease compared to its value when MLD is 198. However, MMN has a significant impact on reconstruction results. With the same number of output faces set to 2592, $p_{2\tau}$ decreases by 9.54% when MMN is 1,

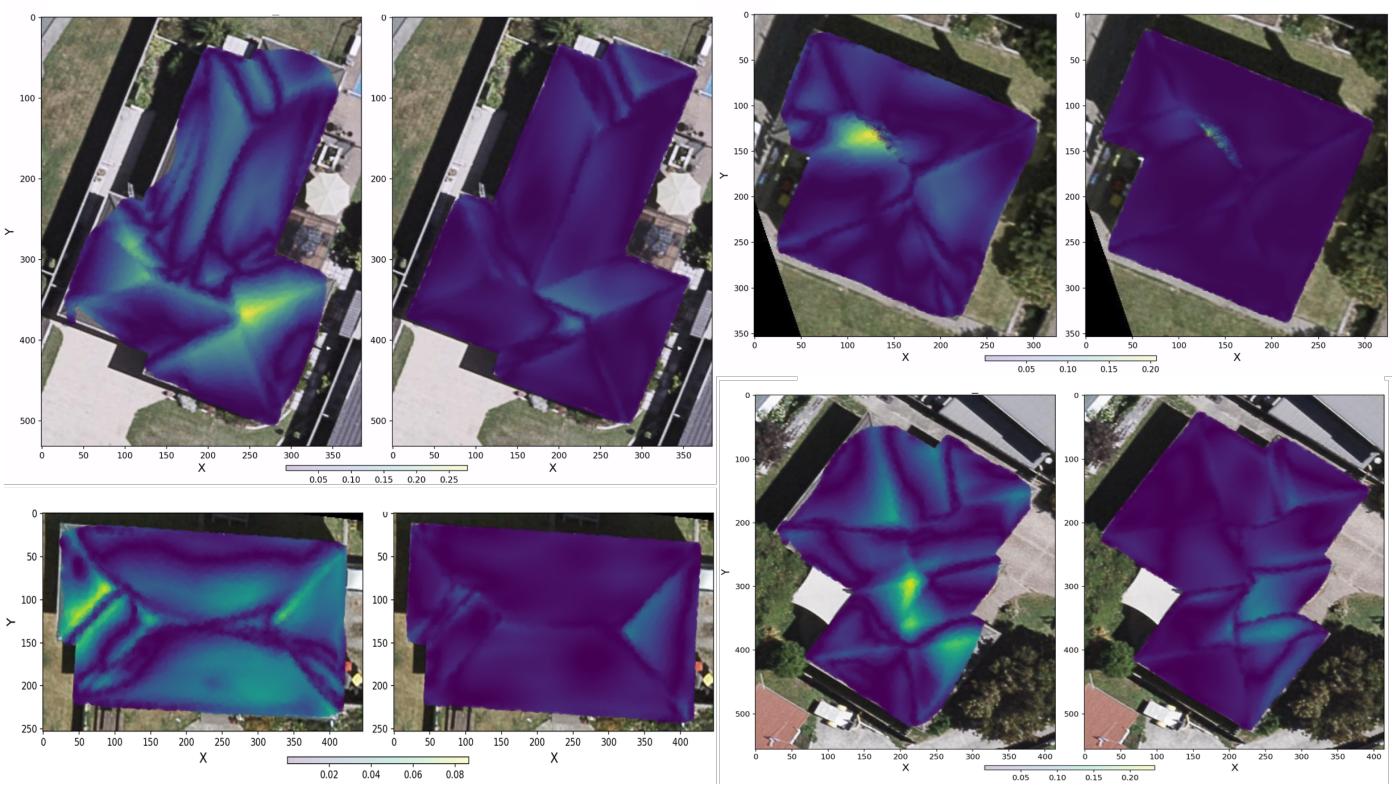


Figure 9: Visualization example of the normalized error along the Z-axis for the results of RooFormer and P2M_plane. Brighter colors indicate higher errors, as shown in the legend..

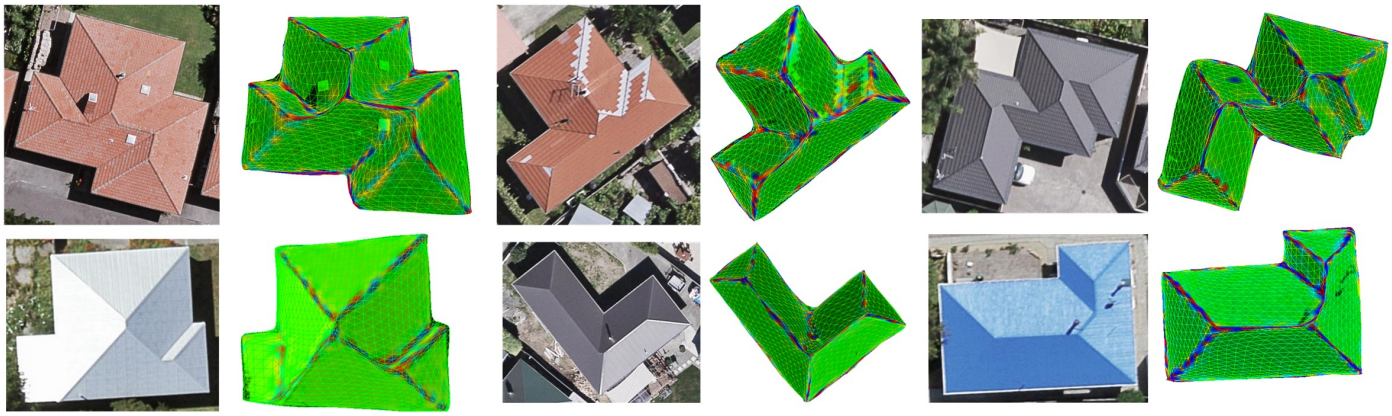


Figure 10: Visualization of the principal directions of curvature of the reconstructed meshes.

Table 2: Performance metrics for different input image sizes

Metrics	112×112	224×224	448×448
Training Time (hours)	3.71	3.95	4.30
Training Throughput (images/s)	16.00	15.92	14.08
Training Memory Usage (GB)	9.43	11.54	17.68
Inference Throughput (images/s)	227.67	223.23	229.31

compared to the result with 3 MMNs. As shown in Figure 13, the reconstructed roofs with 3 MMNs have a more precise geometry and regular boundaries compared to those with 1 MMN.

5.4.2. Ablation studies on at the loss level

To investigate the effects of different loss terms on reconstructed results, we evaluate the function of key loss terms: l_{geom}, l_{mask}, l_{reg}, and the subterms of l_{reg} (i.e., l_{lap} and l_{edge}).

Figure 11: Qualitative results of real-world aerial images from OpenAerialMap online.

Table 3: Table 3: Quantitative comparison using different network architectures

Method	Total Params	d_{cd}	p_τ	$p_{2\tau}$	z_{L1}	z_{L2}
MLH_1	53.6 M	3.84e-4	57.34	82.67	0.0154	0.166
MLH_2	55.7 M	4.11e-4	57.96	83.19	0.0149	0.153
MLD_118	54.0 M	3.85e-4	56.11	81.94	0.0160	0.170
MLD_158	56.6 M	3.74e-4	58.30	83.39	0.0150	0.159
MLN_1	52.8 M	3.93e-4	56.57	82.12	0.0155	0.165
MLN_3	55.7 M	3.72e-4	57.46	83.06	0.0149	0.151
MMN_1	52.4 M	5.03e-4	48.40	74.54	0.0217	0.206
Full_4_198_6_3	58.8 M	3.57e-4	58.88	83.95	0.0140	0.144

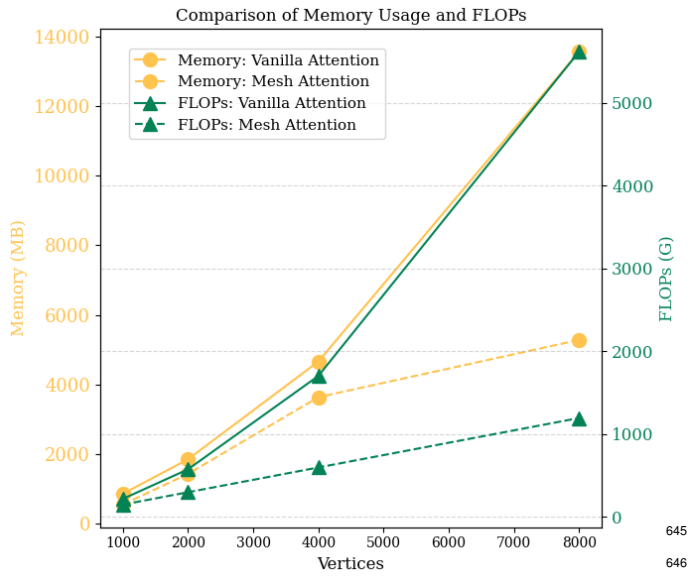


Figure 12: Performance metrics for Mesh Self-Attention and Vanilla Self-Attention.

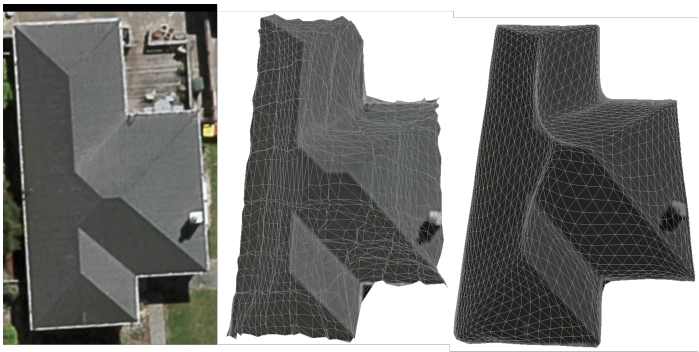


Figure 13: Visualization comparison of the results with MMN=1 and MMN=3 under the same number of output faces.

As shown in Figure 14, removing the mesh regularization term l_{reg} and its subterms l_{lap} and l_{edge} from the loss function results in excessively long and uneven edge lengths of the reconstructed geometric primitives. The issue of excessively long edges caused by the l_{lap} term is more severe than that caused by the l_{edge} term. Removing the mask term l_{mask} from the loss function impairs the accuracy of reconstructed roof meshes at their boundaries, leading to issues such as incomplete roof reconstruction and the inclusion of non-roof areas. This problem is particularly severe when there are objects that interfere with roofs, such as vehicles and shadows, as shown in the red re-

gions of Figure 14. These results highlight the importance of each loss term in contributing to overall performance.

We evaluated three key loss weight coefficients: l_2, l_4, and l_5. Table 4 highlights how changes to these coefficients affect reconstruction results. Selecting optimal values is a combinatorial optimization challenge. The coefficients used in Section 4.1 were derived from extensive experimentation and are recommended for this task.

5.5. Discussion

The proposed network enables the reconstruction of detailed and textured 3D roof meshes from single-view RS images. We have conducted quantitative analysis, qualitative analysis, and ablation studies on the IMP dataset and open aerial images. The following observations are worthwhile for further discussion.

Roof structure and color The proposed network performs well on roofs with various structures, as shown in Figure 8. For low complexity roofs, such as those with fewer than eight roof ridges and hips, the reconstructed meshes exhibit clear and precise structures. However, for roofs with numerous ridges and hips, the reconstructed meshes occasionally show a decline in structural accuracy, particularly in small roof hip areas. This may be due to the smaller regions contributing less significantly to the loss gradient during backpropagation. Additionally, we observed that the proposed method is not sensitive to roof color, producing satisfactory results on roofs of different colors.

Occlusion In RS images, some roofs may be occluded by nearby tree canopies. Theoretically, occluded areas are difficult to reconstruct due to the loss of contextual texture. However, in practice, the proposed method achieves satisfactory reconstruction results in areas mildly occluded by vegetables, as shown in the blue box in Figure 15a. This is primarily because roofs typically have regular shapes, allowing the model to infer missing parts after being trained on extensive data. Similarly, humans can identify the actual roof boundaries from occluded RS images. Nevertheless, as illustrated in the red box in Figure 15b, when the occluded area is substantial, the reconstruction by RooFormer deviates from the actual roof extent. Furthermore, Figure 16 shows the reconstruction results of synthetic images with varying levels of occlusion by vegetables, where it can be observed that significant occlusion along structural lines impacts the reconstruction performance of RooFormer.

Roof appendages As shown in the red box and curvature mapping in Figure 15c, the reconstructed meshes do not include small roof appendages such as chimneys and dormers. There are two main reasons for this. First, the features and geometric structures of tiny objects are difficult to capture. Second, tiny roof appendages are not modeled in the GT meshes used in training, leading the trained model to interpret these appendages' texture features as part of the roof's flat surface. These are areas that need to be optimized in future data sets.

6. Conclusions

As crucial elements of buildings, 3D roof models are essential for various analyses, including solar potential analysis, urban microclimate simulation, and energy efficiency assessments. The diversity and complexity of roof structures has for a long time presented significant challenges for accurate and efficient 3D roof reconstruction. RS images provide wide coverage, frequent updates, and easy public access. However, little attention has been given to the reconstruction of detailed 3D roof models using DNNs from RS images.

In this work, we propose an end-to-end learning framework named RooFormer for reconstructing detailed and textured 3D roof models in mesh format. Given an input high-resolution remote sensing image containing a complete roof, RooFormer can automatically infer a 3D roof model. RooFormer consists of a MaskFormer branch, which identifies and focuses on roof features, and a MeshFormer branch, which predicts detailed roof meshes. In the MeshFormer branch, a local self-attention mechanism is developed to interpret mesh features, and a positional embedding layer is designed to integrate geometric and texture features. In addition, to measure geometric similarity between predicted and GT meshes, the loss function incorporates terms from both image and mesh domains. The proposed geometric loss term, compared to existing 3D metrics like Chamfer distance, more accurately reflects the geometric differences in meshes.

After evaluation and ablation experiments, the height errors of RooFormer are 0.014, approximately one-third those of state-of-the-art methods. Visually, the reconstruction accurately reflects the geometric contours and structures of roofs, even under slight occlusion. We tested the trained RooFormer on publicly available online aerial images and achieved promising results, demonstrating its generalization capability. In addition, we discussed the proposed model in terms of occlusion and roof appendages, roof structure, and color.

We consider this work a step forward for 3D roof reconstruction from RS data, and potentially the basis for 3D building reconstruction from RS imagery. Our framework promises to enable richer building modeling and analysis for broad digital city applications. The framework could be improved from the following perspectives for future work: 1) introducing multi-source data fusion to predict complete 3D building models; 2) extending the framework to predict small roof attachments; 3) exploring the use of additional data, such as DEMs or multi-view imagery. Furthermore, we will explore a network with more parameters based on RooFormer to predict 3D roofs at a national scale.

Acknowledgments

The work was supported by the National Natural Science Foundation of China under Grant 42425108. This research is part of the project Multi-scale Digital Twins for the Urban Environment: From Heartbeats to Cities, which is supported by the Singapore Ministry of Education Academic Research Fund Tier 1. The authors acknowledge financial support from China Scholarship Council. Fundamental Research Funds for the Central Universities under Grant 2042022dx0001.

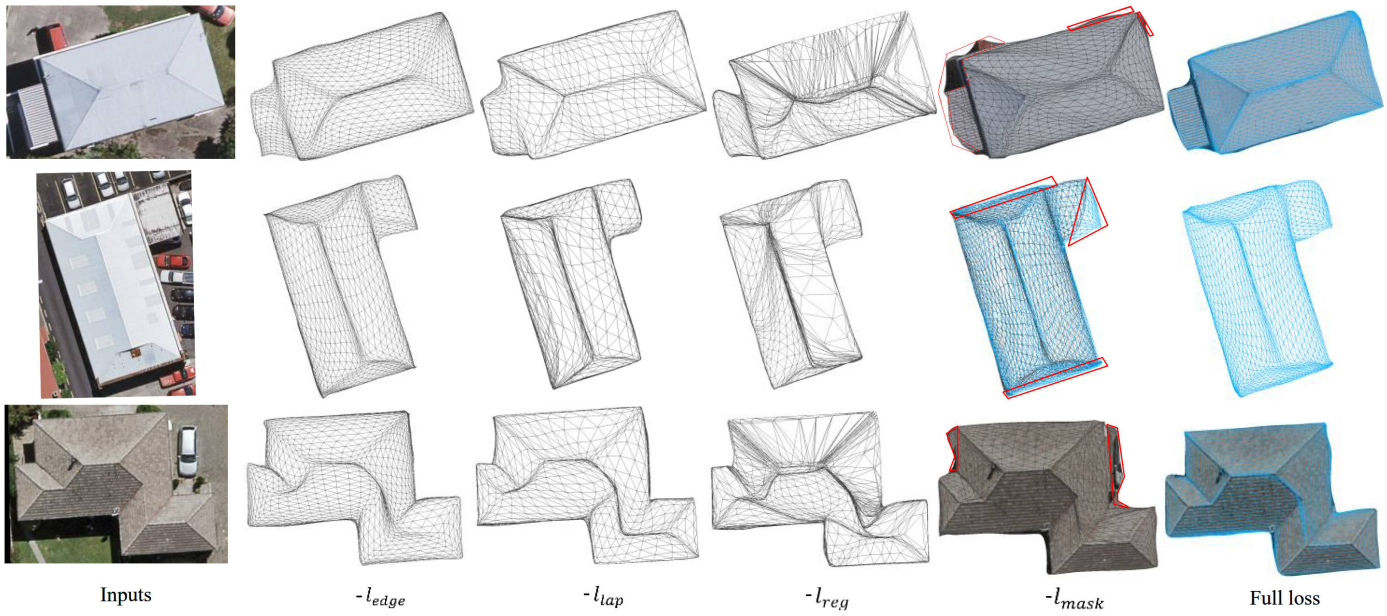


Figure 14: Qualitative results from ablation studies reflecting the contribution of each loss term.

Table 4: Quantitative comparison using different loss weight coefficients

Metrics	Base	λ_2		λ_4		λ_5	
		0.01	1	0.05	5	0.01	1
d_{cd}	3.57e-4	3.93e-4	4.33e-4	3.75e-4	4.26e-4	3.82e-4	3.91e-4
$p\tau$	58.88	58.23	53.07	58.11	57.21	58.18	57.89
$p2\tau$	83.95	83.04	79.14	83.12	82.54	83.35	83.16
z_{L1}	0.0140	0.0143	0.0183	0.0148	0.0161	0.0151	0.0155
z_{L2}	0.1442	0.1509	0.1953	0.1587	0.1814	0.1596	0.1683

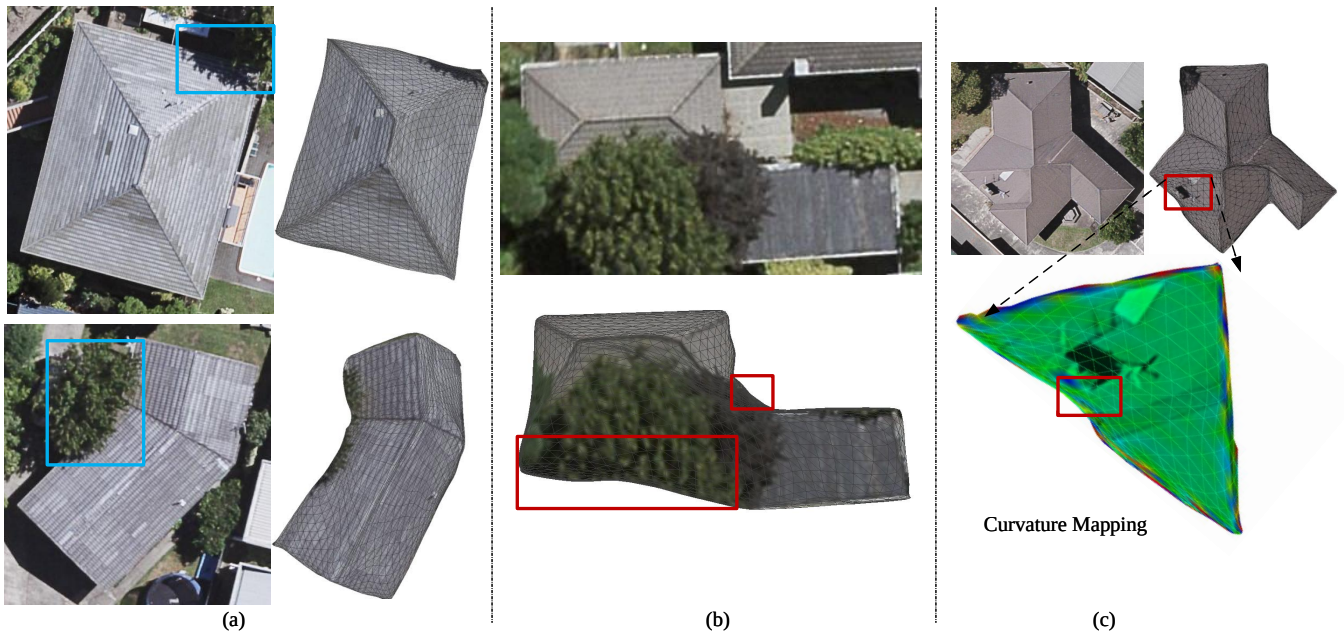


Figure 15: Example results of occlusion and failure cases: (a) Results in mildly occluded areas. (b) Results in largely occluded areas. (c) Results and primary direction of curvature for small roof appendages.

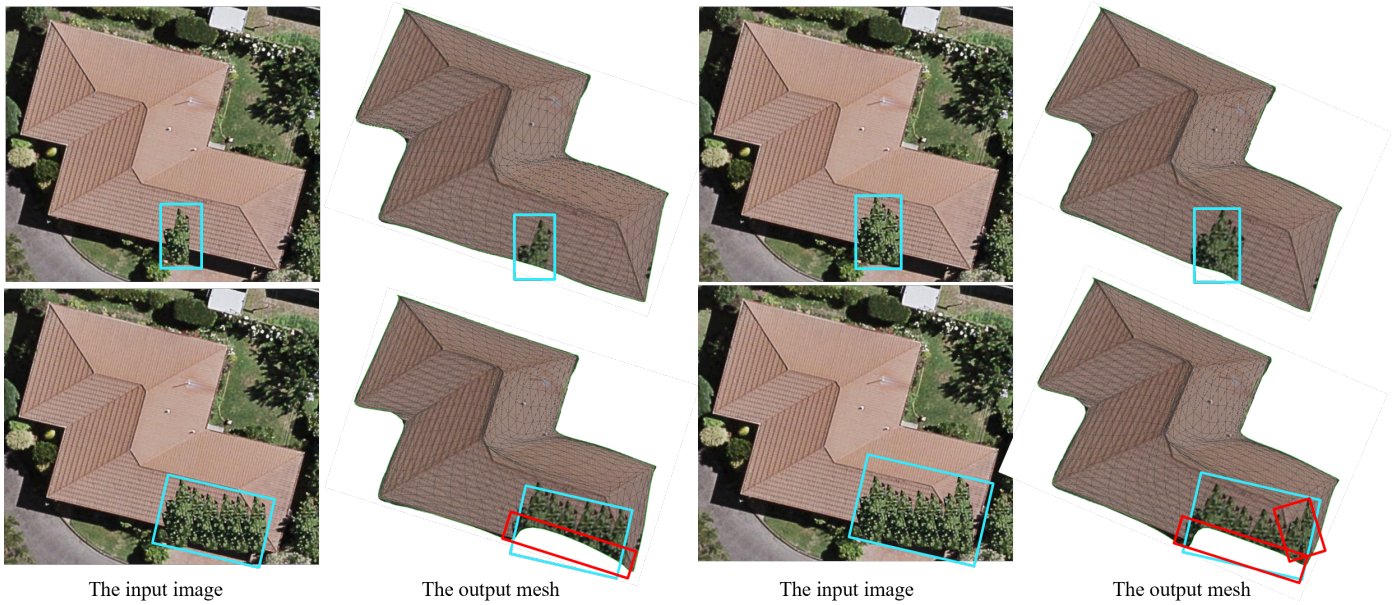


Figure 16: The reconstructed roof meshes under different levels of occlusion by vegetables: the green box indicates areas with artificial occlusion at varying levels, while the red box highlights regions with significant errors.

Data and codes availability statement

The data and codes that support the findings of the present study are available on DOI: 10.6084/m9.figshare.28604519

References

Agugiaro, G., 2016. Energy planning tools and citygml-based 3d virtual city models: experiences from trento (italy). Applied Geomatics 8, 41–56.

Amirkolaee, H.A., Arefi, H., 2019. Height estimation from single aerial images using a deep convolutional encoder-decoder network. ISPRS Journal of Photogrammetry and Remote Sensing 149, 50–66. doi:https://doi.org/10.1016/j.isprsjprs.2019.01.013.

Biljecki, F., Ledoux, H., Stoter, J., 2017. Generating 3d city models without elevation data. Computers, Environment and Urban Systems 64, 1–18. doi:https://doi.org/10.1016/j.compenvurbsys.2017.01.001.

Castrejón, L., Kundu, K., Urtasun, R., Fidler, S., 2017. Annotating object instances with a polygon-rnn. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 4485–4493.

Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F., 2015. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR]. Stanford University — Princeton University — Toyota Technological Institute at Chicago.

Charles, R.Q., Su, H., Kaichun, M., Guibas, L.J., 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 77–85. doi:10.1109/CVPR.2017.16.

Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K.P., Yuille, A.L., 2016. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence 40, 834–848.

Chen, S., Ogawa, Y., Zhao, C., Sekimoto, Y., 2023. Large-scale individual building extraction from open-source satellite imagery via super-resolution-based instance segmentation approach. ISPRS Journal of Photogrammetry and Remote Sensing 195, 129–152. doi:https://doi.org/10.1016/j.isprsjprs.2022.11.006.

Cheng, J., Deng, C., Su, Y., An, Z., Wang, Q., 2024. Methods and datasets on semantic segmentation for unmanned aerial vehicle remote sensing images: A review. ISPRS Journal of Photogrammetry and Remote Sensing 211, 1–34. doi:https://doi.org/10.1016/j.isprsjprs.2024.03.012.

Dehbi, Y., Henn, A., Gröger, G., Stroh, V., Plümer, L., 2021. Robust and fast reconstruction of complex roofs with active sampling from 3d point clouds. Transactions in GIS 25, 112–133. doi:https://doi.org/10.1111/tgis.12659.

Diakogiannis, F.I., Waldner, F., Caccetta, P., Wu, C., 2020. Resunet-a: A deep learning framework for semantic segmentation of remotely sensed data. ISPRS Journal of Photogrammetry and Remote Sensing 162, 94–114. doi:https://doi.org/10.1016/j.isprsjprs.2020.01.013.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations.

Dutta, S., Das, M., 2023. Remote sensing scene classification under scarcity of labelled samples—a survey of the state-of-the-arts. Computers & Geosciences 171, 105295. doi:https://doi.org/10.1016/j.cageo.2022.105295.

Goetz, M., 2013. Towards generating highly detailed 3d citygml models from openstreetmap. International Journal of Geographical Information Science 27, 845–865. doi:10.1080/13658816.2012.721552.

Goetz, M., Zipf, A., 2012. Towards defining a framework for the automatic derivation of 3d citygml models from volunteered geographic information. International Journal of 3-D Information Modeling 1, 16. doi:10.4018/ij3dim.2012040101.

Guo, H., Shi, Q., Du, B., Zhang, L., Wang, D., Ding, H., 2021. Scene-driven multitask parallel attention network for building extraction in high-resolution remote sensing images. IEEE Transactions on Geoscience and Remote Sensing 59, 4287–4306. doi:10.1109/TGRS.2020.3014312.

Hongchao Fan, Alexander Zipf, Q.F., Neis, P., 2014. Quality assessment for building footprints data on openstreetmap. International Journal of Geographical Information Science 28, 700–719. doi:10.1080/13658816.2013.867495.

Huang, X., Zhang, Z., Li, J., 2024. China's first sub-meter building footprints derived by deep learning. Remote Sensing of Environment 311, 114274. doi:https://doi.org/10.1016/j.rse.2024.114274.

Jiang, T., Wang, Y., Zhang, Z., Liu, S., Dai, L., Yang, Y., Jin, X., Zeng, W., 2023. Extracting 3-d structural lines of building from als point clouds using graph neural network embedded with corner information. IEEE Transactions on Geoscience and Remote Sensing 61, 1–28. doi:10.1109/TGRS.2023.3278589.

Kada, M., Mckinley, L., 2009. 3d building reconstruction from lidar based on a cell decomposition approach. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 38, 47–52.

Kölle, M., Laupheimer, D., Schmohl, S., Haala, N., Rottensteiner, F.,

Wegner, J.D., Ledoux, H., 2021. The hessigheim 3d (h3d) benchmark on semantic segmentation of high-resolution 3d point clouds and textured meshes from uav lidar and multi-view-stereo. ISPRS Open Journal of Photogrammetry and Remote Sensing 1, 100001. doi:https://doi.org/10.1016/j.ophoto.2021.100001.

Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., Jia, J., 2022. Stratified transformer for 3d point cloud segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8500–8509.

Lehtola, V.V., Koeva, M., Elberink, S.O., Raposo, P., Virtanen, J.P., Vahdatikhaki, F., Borsci, S., 2022. Digital twin of a city: Review of technology serving city needs. International Journal of Applied Earth Observation and Geoinformation 114, 102915. doi:https://doi.org/10.1016/j.jag.2022.102915.

Lei, B., Liu, P., Milojevic-Dupont, N., Biljecki, F., 2024. Predicting building characteristics at urban scale using graph neural networks and street-level context. Computers, Environment and Urban Systems 111, 102129. doi:10.1016/j.compenvurbsys.2024.102129.

Lei, B., Stouffs, R., Biljecki, F., 2023. Assessing and benchmarking 3D city models. International Journal of Geographical Information Science 37, 788–809. doi:10.1080/13658816.2022.2140808.

Li, W., Meng, L., Wang, J., He, C., Xia, G.S., Lin, D., 2021a. 3d building reconstruction from monocular remote sensing images, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 12528–12537. doi:10.1109/ICCV48922.2021.01232.

Li, W., Zhao, W., Yu, J., Zheng, J., He, C., Fu, H., Lin, D., 2023. Joint semantic geometric learning for polygonal building segmentation from high-resolution remote sensing images. ISPRS Journal of Photogrammetry and Remote Sensing 201, 26–37. URL: https://www.sciencedirect.com/science/article/pii/S0924271623001272, doi:https://doi.org/10.1016/j.isprsjprs.2023.05.010.

Li, X., Wen, C., Hu, Y., Yuan, Z., Zhu, X.X., 2024. Vision-language models in remote sensing: Current progress and future trends. IEEE Geoscience and Remote Sensing Magazine 12, 32–66. doi:10.1109/MGRS.2024.3383473.

Li, Y., Shi, T., Zhang, Y., Chen, W., Wang, Z., Li, H., 2021b. Learning deep semantic segmentation network under multiple weakly-supervised constraints for cross-domain remote sensing image semantic segmentation. ISPRS Journal of Photogrammetry and Remote Sensing 175, 20–33. doi:https://doi.org/10.1016/j.isprsjprs.2021.02.009.

Li, Z., Wegner, J.D., Lucchi, A., 2018. Topological map extraction from overhead images. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) , 1715–1724.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).

Liu, Z., Tang, H., Huang, W., 2022. Building outline delineation from vhr remote sensing images using the convolutional recurrent neural network embedded with line segment information. IEEE Transactions on Geoscience and Remote Sensing 60, 1–13. doi:10.1109/TGRS.2022.3154046.

Madhuanand, L., Nex, F., Yang, M.Y., 2021. Self-supervised monocular depth estimation from oblique uav videos. ISPRS Journal of Photogrammetry and Remote Sensing 176, 1–14. doi:https://doi.org/10.1016/j.isprsjprs.2021.03.024.

Mao, Y., Chen, K., Zhao, L., Chen, W., Tang, D., Liu, W., Wang, Z., Diao, W., Sun, X., Fu, K., 2023. Elevation estimation-driven building 3-d reconstruction from single-view remote sensing imagery. IEEE Transactions on Geoscience and Remote Sensing 61, 1–18. doi:10.1109/TGRS.2023.3266477.

Microsoft, 2024. Global ml building footprints. https://github.com/microsoft/GlobalMLBuildingFootprints. Accessed: 2024-08-10.

Nurkarim, W., Wijayanto, A.W., 2023. Building footprint extraction and counting on very high-resolution satellite imagery using object detection deep learning framework. Earth Sci. Inform. 16, 515–532.

Park, Y., Guldmann, J.M., 2019. Creating 3d city models with building footprints and lidar point cloud classification: A machine learning approach. Computers, Environment and Urban Systems 75, 76–89. doi:https://doi.org/10.1016/j.compenvurbsys.2019.01.004.

Ren, J., Zhang, B., Wu, B., Huang, J., Fan, L., Ovsjanikov, M., Wonka, P., 2021. Intuitive and efficient roof modeling for reconstruction and synthesis. ACM Trans. Graph. 40. doi:10.1145/3478513.3480494.

Riegler, G., Ulusoy, A.O., Geiger, A., 2017. Octnet: Learning deep 3d representations at high resolutions, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6620–6629. doi:10.1109/CVPR.2017.701.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham. pp. 234–241.

Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E., 2015. Multi-view convolutional neural networks for 3d shape recognition, in: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 945–953. doi:10.1109/ICCV.2015.114.

Tatarchenko, M., Park, J., Koltun, V., Zhou, Q.Y., 2018. Tangent convolutions for dense prediction in 3d. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition , 3887–3896.

Vallet, B., Pierrot-Deseilligny, M., Boldo, D., Brédif, M., 2011. Building footprint database improvement for 3d reconstruction: A split and merge approach and its evaluation. ISPRS Journal of Photogrammetry and Remote Sensing 66, 732–742. doi:https://doi.org/10.1016/j.isprsjprs.2011.06.005.

Vandita Srivastava, R.A., George, S.V., 2024. Investigations on extraction of buildings from rs imagery using deep learning models. International Journal of Remote Sensing 45, 68–100. doi:10.1080/01431161.2023.2292016.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I., 2017. Attention is all you need, in: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc.. pp. 5999–6009.

Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G., 2018. Pixel2mesh: Generating 3d mesh models from single rgb images., in: ECCV (11), Springer. pp. 55–71.

Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L., 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 548–558. doi:10.1109/ICCV48922.2021.00061.

Yang, G., Xue, F., Zhang, Q., Xie, K., Fu, C.W., Huang, H., 2023. Urbanbis: a large-scale benchmark for fine-grained urban building instance segmentation, in: ACM SIGGRAPH 2023 Conference Proceedings, Association for Computing Machinery, New York, NY, USA. doi:10.1145/3588432.3591508.

Yu, D., He, L., Ye, F., Jiang, L., Zhang, C., Fang, Z., Liang, Z., 2022. Unsupervised ground filtering of airborne-based 3d meshes using a robust cloth simulation. International Journal of Applied Earth Observation and Geoinformation 111, 102830. doi:https://doi.org/10.1016/j.jag.2022.102830.

Yu, D., Yue, P., Wu, B., Biljecki, F., Chen, M., Lu, L., 2024. Towards an integrated approach for managing and streaming 3d spatial data at the component level in spatial data infrastructures. International Journal of Geographical Information Science 0, 1–25. doi:10.1080/13658816.2024.2434606.

Yu, D., Yue, P., Ye, F., Tapete, D., Liang, Z., 2023. Bidirectionally greedy framework for unsupervised 3d building extraction from airborne-based 3d meshes. Automation in Construction 152, 104917. doi:https://doi.org/10.1016/j.autcon.2023.104917.

Yuan, Y., Tang, J., Zou, Z., 2021. Vanet: a view attention guided network for 3d reconstruction from single and multi-view images, in: 2021 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. doi:10.1109/ICME51207.2021.9428171.

Zhang, C., Fan, H., Kong, G., 2021. Vgi3d: an interactive and low-cost solution for 3d building modelling from street-level vgi images. Journal of Geovisualization and Spatial Analysis 5, 18. doi:10.1007/s41651-021-00086-7.

Zhang, Y., Ren, J., Pu, Y., Wang, P., 2020. Solar energy potential assessment: A framework to integrate geographic, technological, and economic indices for a potential analysis. Renewable Energy 149, 577–586. doi:https://doi.org/10.1016/j.renene.2019.12.071.

Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V., 2021. Point transformer, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 16259–16268.

Zhao, W., Persello, C., Stein, A., 2022. Extracting planar roof structures from very high resolution images using graph neural networks. ISPRS Journal of Photogrammetry and Remote Sensing 187, 34–45. doi:https://doi.org/10.1016/j.isprsjprs.2022.02.022.

Zhu, L., Wang, X., Ke, Z., Zhang, W., Lau, R., 2023. Biformer: Vision transformer with bi-level routing attention, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10323–10333. doi:10.1109/CVPR52729.2023.00995.