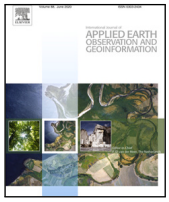
International Journal of Applied Earth Observation and Geoinformation 133 (2024) 104096

Contents lists available at ScienceDirect

International Journal of Applied Earth Observation and Geoinformation

journal homepage: www.elsevier.com/locate/jag



Roof plane parsing towards LoD-2.2 building reconstruction based on joint learning using remote sensing images

Yajin Xu [a,*], Juilson Jubanski [a], Ksenia Bittner [b], Florian Siegert [a,c]

[a] 3D RealityMaps GmbH, Landsberger Straße 314, Munich, 80687, Bavaria, Germany
[b] German Aerospace Center (DLR), Münchener Straße 20, Weßling, 82234, Bavaria, Germany
[c] GeoBio–Center, Ludwig Maximilian University of Munich, Richard–Wagner–Straße 10, Munich, 80333, Bavaria, Germany

ARTICLE INFO

Keywords:
Building reconstruction
Deep learning
Data fusion
Remote sensing
Computer vision

ABSTRACT

Building models are important for urban studies. Remote sensing multi-spectral (MS) images are widely used for its rich semantic information. The lack of geometry features is fulfilled by introducing photogrammetry derived digital surface models (DSMs), resulting in pairs of DSMs and MS images. Utilizing such pairs and a convolutional neural network, level of detail (LoD) 2.2 building models, which contain roof planes and major roof elements (e.g. dormers), are reconstructed in this work. Leveraging both raster and vector predictions, 3-D building models with straight edges and sharp corners are obtained. The proposed two-stage method first extracts vectorized roof lines from pairs of DSMs and RGB images, followed by generation of detailed 2-D and 3-D polygonal building models. We conducted our experiments based on two datasets: a custom dataset in Landsberg am Lech in Germany, and an open dataset named Roof3D. For the custom dataset, our proposed model achieved mean average precision (mAP) for building roof vertices of 64.3% and for building roof lines of 54.5% at highest. Mean precision and recall for reconstructed 2-D building roof plane polygons are 52.2% and 54.7% respectively. For the Roof3D dataset, mAP is reported to be 25.3% and 12.4% for the extracted building roof lines and roof plane polygons respectively.

1. Introduction

Detailed building models are useful in many urban studies, e.g. energy budget estimation, real estate valuation. Remote sensing technique has become a major source of building reconstruction (Brenner, 2005), especially with multi-spectral (MS) imaging, for its rich spectral information. With the rise of deep learning methods, convolutional neural networks (CNNs) are comprehensively studied and applied to the task of building reconstruction, in sense of RGB images (Mahmud et al., 2020; Robinson et al., 2022).

Despite of their rich spectral features, the lack of geometry information of RGB images leads to final products with compromised quality. Therefore, to achieve better performance, the combination of RGB images and geometric information is inevitable. Photogrammetry derived digital surface model (DSM) and corresponding orthophoto form naturally an ideal image pair for 3-D building extraction, consisting both spectral and geometric information.

Although there are many conventional methods targeting at building reconstruction using DSMs and RGB image pairs (Arefi and Reinartz, 2013; Mousa et al., 2019; Liu et al., 2021), limited amount of deep learning based fusion of DSMs and RGB images exist in literature. Some pioneering works integrating DSMs and RGB images in neural networks are applied to building boundary extraction (Bittner et al., 2018) and to roof planes extraction (Schuegraf et al., 2024). However, these methods present curved building corners in the final products and lack of finer details of roof objects.

Depending on the degree of reconstructed details, Biljecki et al. (2016) proposed different level of detail (LoD) building models. In our work, we adapt similar definitions and aim to extract LoD-2.2 models using remote sensing images. Specifically, the LoD-2.2 models under study consist of individual roof planes, dormer surfaces and other large roof installations (larger than 4 m^2), with assumed vertical walls and no overhangs. There are some recent studies trying to extract 2-D vectorized LoD-2 models using remote sensing images (Hensel et al., 2021; Qian et al., 2022; Zhao et al., 2022), but the authors focused only on roof lines extraction. For down-stream tasks, complete and closed building polygons are needed, especially for 3-D model reconstruction, as well as higher LoDs.

Based on the above mentioned research gaps, we propose a deep learning method that combines DSMs and RGB images for LoD-2.2

* Corresponding author.
E-mail addresses: xu@realitymaps.de (Y. Xu), jubanski@realitymaps.de (J. Jubanski), ksenia.bittner@dlr.de (K. Bittner), siegert@realitymaps.de (F. Siegert).

https://doi.org/10.1016/j.jag.2024.104096
Received 18 June 2024; Received in revised form 25 July 2024; Accepted 10 August 2024
Available online 22 August 2024

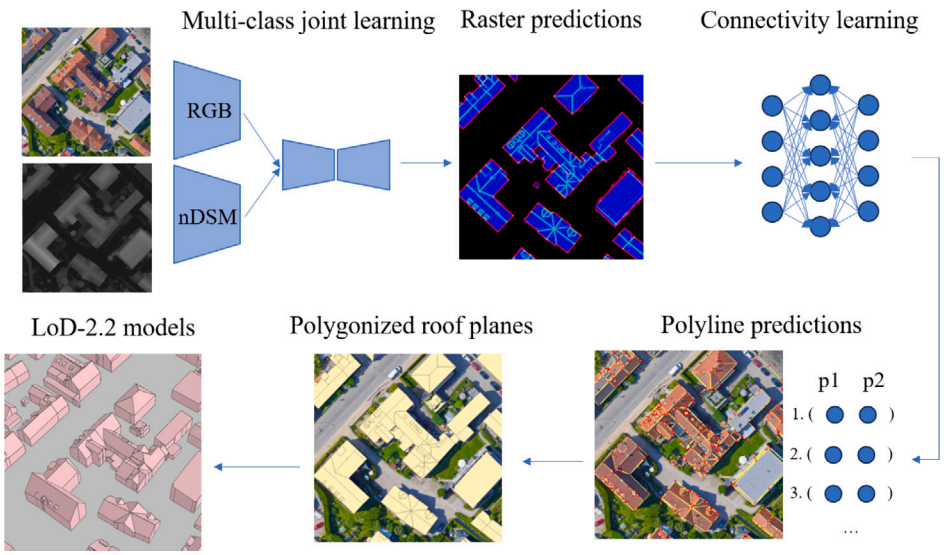


Fig. 1. General workflow of our proposed method. Our designed CNN takes as input RGB image and nDSM and predicts the geometry primitives according to our proposed building representation model. Roof lines are extracted by the connectivity learning module, and polygons are generated based on the predicted roof lines. LoD-2.2 models are finally built using 2-D roof plane polygons.

building reconstruction. More specifically, our targeted task is LoD-2.2 3-D building reconstruction using paired RGB images and DSMs. Our work contributes to the study of paired DSMs and orthophotos for building extraction, and fills the gap in literature from roof lines extraction to 3-D building models generation.

Fig. 1 presents our proposed method. We propose a multi-class joint learning backbone network to first obtain raster predictions. Our designed network is able to fuse information from RGB images and DSMs to improve model performance. More specifically, we use normalized DSM (nDSM) for better generalization, which is calculated by subtracting digital terrain model (DTM) from a DSM. The following connectivity learning converts the raster predictions to vectors of straight lines. Using the raster and vector predictions, 2-D roof plane polygons are generated, and are extruded to 3-D polygons based on DSMs. With added vertical walls and projected ground polygons, final 3-D building models are obtained in polygonal format. In all, our contributions are summarized as follows:

- we propose a backbone network that fuses geometric and spectral information for detailed building extraction, contributing to the study of DSM and orthophoto pairs,
- our work fills the gap between predictions of roof lines and 3-D building models, with intermediate products being closed roof plane polygons with straight edges and sharp corners,
- we propose a simple yet effective LoD-2.2 building abstraction as semantic raster prediction which models all kinds of buildings,
- we develop an effective workflow of generating LoD-2.2 3-D building models in polygonal format based on RGB images and DSMs.

2. Related work

2.1. Building footprint extraction

There are a lot of emerging deep learning based methods for the task of 2-D building footprint extraction in literature. Most of research use geometry primitives as base information to reconstruct building footprints. Three types of geometry primitives are usually involved: segments, lines and points.

The most straightforward method to extract building footprint is to apply semantic/instance segmentation models. This kind of methods is built upon a backbone network to extract building segments (Mahmud et al., 2020; Robinson et al., 2022), followed by post-processing such as regularization (Zhao et al., 2018; Zorzi et al., 2021). Other than building segments, building lines and corners are also useful. Under this category of methods, building corners and their connections are predicted (Zorzi et al., 2022), sequence of building corners are generated (Li et al., 2019; Huang et al., 2022), or building edges are incorporated in neural network design (Chen et al., 2022).

These methods provide decent building outlines and/or building polygons, but the final products are not detailed enough semantically. Besides, they cannot be applied directly to the task of roof planes extraction, for they generate one polygon for each building instance.

2.2. Building roof planes extraction

Due to highly varying and complicated building roof structures, the task of building roof planes extraction is much more challenging than building outline or building footprint extraction. In literature, the extraction is mostly based on roof lines and building corners in context of MS images.

Based on edge mask predictions, Qian et al. (2022) used semantic segmentation to predict rasterized roof lines, but without further polygonization for closed roof plane polygons, which limits its application. Additionally, the predicted rasterized roof lines suffer heavily from problems such as broken or irregular line segments, leading to rounded and curved building corners (Schuegraf et al., 2024).

Based on the issues mentioned above, instead of rasterized predictions, it is beneficial to switch to vector learning. The idea is adapted widely in line parsing and wireframe parsing models (Huang et al., 2018; Zhou et al., 2019), where vertices and their connections are predicted. This kind of methods has the advantage of generating vector format data and is more semantically abstract and more representative than raster-based methods.

With this idea, Hensel et al. (2021) leveraged a point-pair graph to extract roof line segments. Similarly, Zhao et al. (2022) used a graph convolutional network on the point-pair graph to predict roof lines. These methods produce closed roof plane polygons only when roof lines are without gaps. However, for down-stream tasks, generation of polygons and 3-D models need to be further studied.

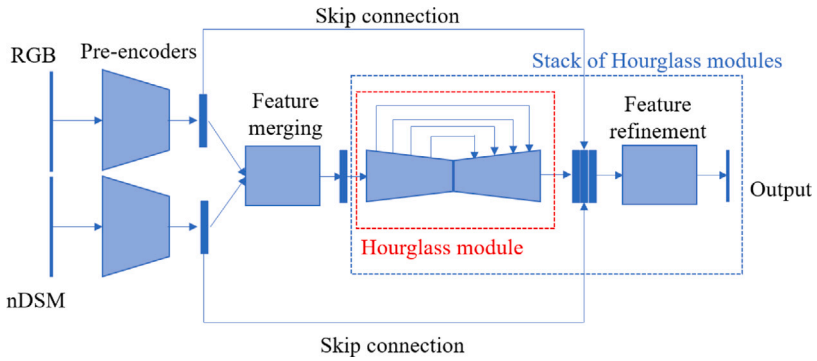


Fig. 2. Proposed network architecture. Our designed backbone network leverages stacked Hourglass modules, and fuses information from RGB and nDSM branches. Skip connection for separately encoded features from the two branches is able to enhance the final extracted features.

2.3. Inclusion of digital surface models

The study of deep learning based fusion of DSMs and orthophotos is still limited in literature. A pioneering work (Bittner et al., 2018) studies the fusion of multi-modal data including nDSM and RGB images at bottle-neck of a CNN, but is applied to building footprint extraction only. Recently, Schuegraf et al. (2024) combined features from RGB images and DSMs for LoD-2 roof planes extraction. These methods exhibit shortcomings of raster prediction based problems, especially the lack of sharp building corners.

In comparison to the existing methods, our method combines the advantages from raster and vector predictions, i.e. rich semantic information and more abstract geometry features, by fusing spectral and geometric information. Consequently, 3-D building models of higher quality with straight edges and sharp corners are obtained.

3. Methodology

Overall, our method consists of two stages to reconstruct 3-D building models. The first stage utilizes a CNN with inputs being paired RGB images and DSMs or nDSMs and outputs being 2-D roof plane polygons. The second part extrudes the extracted roof planes to 3-D polygons and builds complete 3-D building models in polygonal format.

3.1. Backbone network

The base backbone network we chose is stacked Hourglass network (Newell et al., 2016). Its architecture suits well, given that we seek to combine RGB and height information, although the choice of backbone network is not strictly limited. As elevation input, nDSM is preferably used for its better generalization without influences from varying ground heights. The architecture of our design is shown in Fig. 2.

Information from RGB focuses more on texture, while nDSM focuses more on geometry. In our design, we make use of the general idea of stacked Hourglass network, and pre-encode RGB and nDSM differently to achieve information fusion.

The stacked Hourglass network first downsamples and aggregates local information gradually. We follow this procedure and add a second branch to process nDSM in a similar but separate way, namely "pre-encoding". Next, the two sets of feature maps from RGB and nDSM are concatenated and passed to a merging layer, which consists of several Conv-BatchNorm-ReLU stacks. Consequently, the merged feature maps contain both information from RGB and nDSM, and are further refined in the following Hourglass modules.

To better incorporate initial states of the extracted feature maps, we add another skip connection (implemented as concatenation) between the refined features and the pre-encoded features from the two branches. This skip connection allows the network better exploit the separately aggregated features that have different focuses without further refinement or interactions, i.e. image texture and object geometry for RGB and nDSM respectively.

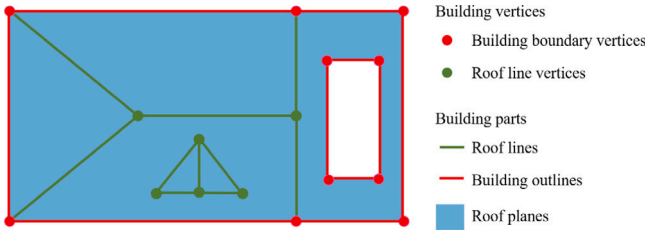


Fig. 3. Our defined building representation model. In general, we divide the roof structures into three basic geometry primitives: segments, lines and points. Specifically, two sets of elements are defined: building vertices and building parts. Further breakdowns are defined based on location relative to the building outline. See main text for details.

3.2. Geometry primitives prediction design

As semantic prediction design (Fig. 3), we define three major types of geometry primitives: building footprint, roof lines and building vertices. Since roof lines at building boundary and inside building footprint have different image texture and geometric features, we further decompose these primitives into outline-related (directly adjacent to non-building areas) and inner-plane-related. For building vertices, we divide them into two classes: building boundary vertices and roof line vertices, all being endpoints of lines. Similarly, for building roof lines, we define building outlines and roof lines. Note that for buildings that contain "holes", e.g. inner yards, we define the corresponding roof lines as building outlines.

We only distinguish different types of building vertices and roof lines based on location for simplicity and efficiency. Moreover, this modeling covers all possible kinds of buildings, even when the building structure is too complicated to be described by a standard eave-ridge representation. One special case is for round-shaped buildings, where the outline is discretized into line segments.

With this building model, we treat the geometry primitives prediction as a problem of classification, and separate the predictions into two parts: building vertices and building parts. Building vertices contain building boundary vertices and roof line vertices. Building parts contain three classes: roof lines, building outlines and roof planes.

3.2.1. Building vertices

For building vertices, we use the same heatmap representation proposed by Zhou et al. (2019). With given image $\mathbf{I}$ and corresponding building vertices $\mathbf{V}$, we first divide the input image $\mathbf{I}$ of size $H \times W$ into $H_b \times W_b$ bins, i.e. $(H_b, W_b) = \left(\lceil \frac{H}{b} \rceil, \lceil \frac{W}{b} \rceil\right)$, where b is the downsize factor. Each pixel in downsized $\mathbf{I}$ is assigned 1 when there is a building vertex present, and 0 otherwise. We use two channels for building vertex heatmaps for the two building vertex classes, denoted as $\mathbf{I}_v \in \mathbb{Z}_{\in\{0,1\}}^{2\times H_b \times W_b}$.

In order to compensate offsets within each bin/pixel, a 2-channel offset map is constructed as $\mathbf{I}_o = (\mathbf{I}_{o,1}, \mathbf{I}_{o,2})$, where

$$\mathbf{I}_{o,1} = \frac{(c_i - v_i)}{b},\ \mathbf{I}_{o,2} = \frac{(c_j - v_j)}{b},$$

where $\mathbf{c} = (c_i, c_j)$ is the location of each bin center in $\mathbf{I}$, and $\mathbf{v} = (v_i, v_j)$ is the location of each building vertex (regardless of vertex class) in $\mathbf{V}$.

3.2.2. Building parts

Building parts contain classes of roof lines, building outlines and roof planes. We construct a 3-channel one-hot-encoded heatmap as building parts representation, denoted as $\mathbf{I}_p \in \mathbb{Z}_{\in\{0,1\}}^{3\times H_b\times W_b}$, using the same downsize factor as building vertices. Since the three classes are exclusive to each other, we are able to make the prediction in a multi-class manner. The building outlines $\mathbf{I}_{p,1}$ and roof lines $\mathbf{I}_{p,2}$ are rasterized with a buffer of 1 pixel on both sides to compensate possible small annotation shifts. The roof planes are obtained with

$$\mathbf{I}_{p,3} = \mathbf{I}_f - (\mathbf{I}_{p,1} + \mathbf{I}_{p,2}),$$

where $\mathbf{I}_f$ is the rasterized building footprint heatmap.

3.3. Multi-class prediction

The network outputs a feature map $\mathbf{F} \in \mathbb{R}^{d\times H_b\times W_b}$, where d is the number of channels. The following prediction head consists of three parallel modules containing Conv-ReLU-Conv layers with different numbers of output channels and activation functions, handling predictions of building vertices, building parts and vertex offsets.

The output logits of building vertices (with three channels) and building parts (with four channels) are passed to a softmax layer, while the vertex offsets logits (with two channels) are input to a sigmoid activation layer added with offset −0.5.

Cross-entropy loss is used for supervision of building vertices and building parts predictions as $\mathcal{L}_v$ and $\mathcal{L}_p$. Masked L_1 loss is used for vertex offsets as

$$L_1 = |\hat{\mathbf{I}}_o - \mathbf{I}_o| \cdot (\mathbf{I}_{v,1} + \mathbf{I}_{v,2}), \tag{1}$$

where $\hat{\mathbf{I}}_o$ is the predicted vertex offset map. Note that we use hat notation to denote predictions associated with corresponding references.

Vertex offsets loss $\mathcal{L}_o$ is then calculated as the sum of L_1 averaged over the total number of reference vertices.

3.4. Connectivity learning

To output data in vector format, we extract building vertices and reconstruct the connections. The building vertex candidates are extracted using the predicted building vertex probability map $\hat{\mathbf{I}}'_v = \hat{\mathbf{I}}_{v,1} + \hat{\mathbf{I}}_{v,2}$. K candidates with highest predicted probability in each image are selected regardless of vertex class, resulting in a predicted building vertices set $\hat{\mathbf{V}}$.

The initial building roof line candidates set $\hat{\mathbf{E}}$ is built upon the building vertex candidates set $\hat{\mathbf{V}}$ by pairing vertex candidates with each other without direction, i.e.

$$\hat{\mathbf{E}} = \{\hat{l} = (\hat{v}_i, \hat{v}_j)\},$$

where

$$i \in \left\{1,\ ...,\ N\left(\hat{\mathbf{V}}\right) - 1\right\}, j \in \left\{i+1,\ ...,\ N\left(\hat{\mathbf{V}}\right)\right\},$$

where $N(\cdot)$ is the total number of elements.

Labels of connections for building roof line candidates are assigned by matching vertex candidates with reference vertices from $\mathbf{V}$ based on Euclidean distance. A building vertex candidate is considered matched if the distance to the closest reference vertex is smaller than a threshold θ_v. For each line candidate, we assign label 1 indicating positive connection only when its two vertices are both matched with two different reference vertices and are connected in reference, otherwise we assign label 0. We keep all positive connections, and sample the same number of negative connections to balance training samples, resulting in a subset $\tilde{\mathbf{E}}$ of $\hat{\mathbf{E}}$. An example illustration is provided in Fig. 4.

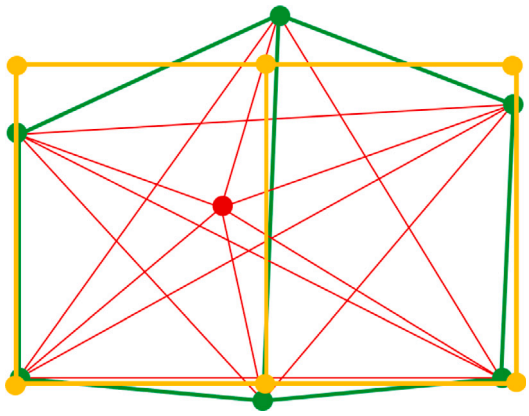


Fig. 4. An illustration of line candidate pairing. The yellow points and lines simulate reference building vertices and building roof lines. The green points simulate matched predicted building vertices, and the red point simulates an unmatched predicted building vertex. The red and green lines simulate initial building roof line candidates obtained by pairing predicted building vertices, with red and green marking negative and positive connections respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Feature vectors are constructed by sampling n_1 points along each line in $\tilde{\mathbf{E}}$. Features at sampled locations in $\mathbf{F}$ are recorded, referred as line of interest (LoI) pooling (Zhou et al., 2019). Different from the existing model, we sample points in a buffered area around each line candidate. The reasoning behind is that a building roof line appears never as a line segment with width 1 in image but a rectangular region with certain width. Considering a buffered area leads to a more robust feature representation for each line candidate. Additionally, it helps compensate possible small misalignment between connected lines and input image.

In practice, we calculate two parallel lines for each line candidate on both sides with offset ±0.5 and ±1, and sample along each added line n_1 locations, resulting in total $5 \times n_1$ sampled points. The feature vectors at each sampled location are concatenated together to form a feature matrix $\mathbf{f} \in \mathbb{R}^{5\times n_1\times d}$. Max-pooling is used to reduce the feature matrix into shape $n_2 \times d$, which is flattened to obtain feature vectors for each line candidate.

Next, the feature vectors for each line are passed to a multi-layer perceptron (MLP), which outputs the connection probability after a sigmoid activation layer, denoted as $s_i, i \in \{1, \ldots, N(\tilde{\mathbf{E}})\}$. We use binary cross-entropy loss for connectivity learning supervision as $\mathcal{L}_c$.

The whole network is trained in an end-to-end manner, with total loss

$$\mathcal{L} = \mathbf{w}_L \cdot (\mathcal{L}_v, \mathcal{L}_p, \mathcal{L}_o, \mathcal{L}_c)^T, \tag{2}$$

where $\mathbf{w}_L$ is a weight vector containing empirically set balance factors for each loss.

3.5. Further modifications

In connectivity learning, DSMs or nDSMs provide useful geometry information, since building outlines are usually with drastic change of height between two sides, and large height difference along a line usually indicates negative connection. Therefore, we concatenate the input height images as additional feature channels with the extracted feature maps in LoI pooling to improve connectivity learning.

We found it beneficial to use non-directed LoI in experiments. To achieve this, we randomly change the order of start- and endpoints for each building line candidate before LoI pooling. With this change, the connectivity learning has the advantage of input order invariance.

3.6. 3-D polygonization

The 3-D building models studied in this work is in wireframe representation, i.e. 3-D coordinates of building vertices are recorded. Starting from the predicted building roof lines, we build such 3-D models by first generating 2-D polygons of roof planes, and then lift the 2-D polygons to 3-D using random sample consensus (RANSAC) (Fischler and Bolles, 1981).

After thresholding on prediction scores, predicted building roof lines are not all connected with each other. To obtain closed polygons for buildings in cases with dangle points, we design an algorithm to collect and expand the predicted building roof lines set, described in Pseudo-Code 1.

Pseudo-code 1 Closing building roof planes

$\hat{\mathbf{E}} = \left\{\hat{l}_i, s_i\right\}, i \in \left\{1, ..., N\left(\hat{\mathbf{E}}\right)\right\}$ if $s_i \geq \theta_s$
$\hat{I}_{planes} = \left(\text{argmax } \hat{I}_p = 3\right) \wedge \left(\text{RV}\left(\hat{\mathbf{E}}\right) = 0\right)$
$\hat{\mathbf{P}} = \varnothing$
for R in CCA $\left(\hat{I}_{planes}\right)$ **do**
  $\bar{\mathbf{E}} = \hat{\mathbf{E}} \cap \text{binary\_dilation}\,(R)$
  $\tilde{\mathbf{P}}, \bar{\mathbf{E}} = \text{detect\_closed\_polygons}\left(\bar{\mathbf{E}}\right)$
  if $\sum_{\tilde{P} \in \tilde{\mathbf{P}}} A_{\tilde{P}} < A_R$ or $\bar{\mathbf{E}} \neq \varnothing$ **then**
    $\tilde{R} = (R > 0) \wedge \left(\text{RV}\left(\tilde{\mathbf{P}}\right) = 0\right) \wedge \left(\text{RV}\left(\bar{\mathbf{E}}\right) = 0\right)$
    $\tilde{\mathbf{P}} = \tilde{\mathbf{P}} \cup \text{VR}\left(\tilde{R}\right)$
  end if
  $\hat{\mathbf{P}} = \hat{\mathbf{P}} \cup \tilde{\mathbf{P}}$
end for

First, we threshold the predicted connection scores s_i for each predicted building roof line $\hat{l}_i$, and obtain a new line candidates set. Next, roof plane segments are extracted using the predicted building parts map $\hat{I}_p$, where the score in roof planes channel is maximal, i.e. argmax of $\hat{I}_p$ along channel dimension equals to 3.

The extracted roof plane segments could still be connected, resulting in merged roof planes. Therefore, we include the vector predictions by removing predicted building roof lines RV $\left(\hat{\mathbf{E}}\right)$ from the roof plane segments to further separate connected roof planes. Here and in the following, we use RV (rasterize vectors) to denote the algorithm to rasterize vector data, where lines or polygons are rasterized into binary images. All rasterized lines have width of 3.

Each roof plane need to be processed individually. Single roof plane segment is obtained using connected component analysis (CCA) on image $\hat{I}_{planes}$, and the corresponding predicted building roof lines are collected by an intersection check: if a line intersects with a roof plane segment, this line is then assigned to this roof plane segment. For each group of predicted building roof lines, closed linear rings are detected (algorithm detect_closed_polygons in Pseudo-Code 1) to form primitive predicted polygons $\tilde{\mathbf{P}}$, with remaining lines set $\bar{\mathbf{E}}$.

Due to false negative samples, $\tilde{\mathbf{P}}$ might not cover the whole roof plane segment, i.e. sum of the areas of the primitive predicted polygons $\sum_{\tilde{P} \in \tilde{\mathbf{P}}} A_{\tilde{P}}$ is smaller than the area of the roof plane segment A_R, or there are extra lines that lie inside the primitive polygons. Therefore, we use the roof plane segment itself to help complete the missing polygons. The missing areas $\tilde{R}$ contain pixels that are (a) inside the roof plane segment, (b) not in primitive polygons $\tilde{\mathbf{P}}$ and (c) not covered by predicted building roof lines. The missing polygons are then obtained by tracing the contour of each connected component in $\tilde{R}$ followed by Douglas–Peucker algorithm (Douglas and Peucker, 1973), denoted as VR (vectorize rasters) in Pseudo-Code 1. The new polygons as well as the primitive polygons are all recorded, resulting in the final predicted roof plane polygons set $\hat{\mathbf{P}}$. An example of before and after the closing algorithm is shown in Fig. 5.

RANSAC (Fischler and Bolles, 1981) is adopted to obtain 3-D building models in an iterative manner. With given DSM, for each predicted roof plane polygon in $\hat{\mathbf{P}}$, all pixels inside this polygon are selected as input points with coordinates (X, Y, Z). In each iteration, noncolinear sample points are randomly selected from input points, and are used to estimate a flat plane using singular value decomposition (Klasing et al., 2009). The best approximation is selected as the estimation with the most point inliers. Vertical walls are added for each polygon edge, assuming that each wall is flat and vertical. Ground polygons are obtained by projecting the 3-D roof planes to the input DSM. The final 3-D building models are the ensemble of roof planes, walls and grounds, as an example shown in Fig. 6.

4. Dataset and experiment

4.1. Studied datasets

In this work, we used two datasets to test our proposed method. The first dataset studied is a custom dataset. The study area was chosen to be Landsberg am Lech in Germany. Aerial images of ground sampling distance (GSD) 0.05 m were collected. Orthorectified images and corresponding DSMs were calculated using software SURE from nFrames.[1] We used the DTMs published by authorities available online[2] and calculated nDSMs by subtracting DTMs from DSMs.

Reference annotations were collected manually. LoD-2.2 related roof elements were labeled. We aim to cover all possible types of buildings in our study area in annotation. Non-overlapping training and testing areas were selected randomly, with their distributions shown in Fig. 7. We tiled the image into patches of size 512×512 pixels with overlap 256 pixels on four sides. In total, we have training samples 1,379 and testing samples 732.

The second dataset is named Roof3D (Schuegraf et al., 2023). The training data in Roof3D dataset contains RGB satellite images and synthetic images, with paired DSM data, while the validation data are satellite images and corresponding DSMs. The GSD of the satellite images and DSMs are 0.3 m, with size 512×512 pixels. In total, 3,337 training samples and 36 validation samples were used. We report the results on the validation set as our test results.

The annotations in Roof3D are in LoD-2, i.e. excluding roof elements, and are given in format as binary images. To generate the training references for our proposed network, we polygonized the binary masks of roof planes followed by Douglas–Peucker algorithm (Douglas and Peucker, 1973) to obtain simplified roof plane polygons (same algorithm as VR in Pseudo-Code 1). Straight lines were extracted from the roof plane polygons and were used as training references. Polygon vertices were used as reference building vertices. Note that these references are considered as pseudo-labels, since the remaining vertices in polygons after simplification are not all true building vertices. Consequently, overall performance is expected to be compromised.

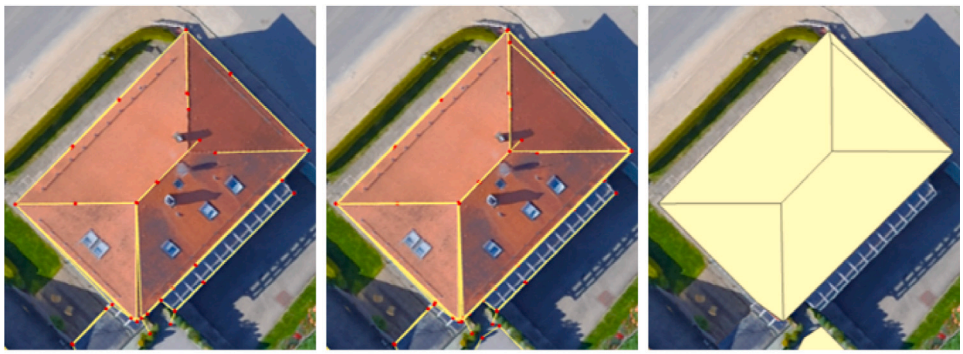
4.2. Implementation and experiments

In our experiments, we fixed the number of stacked Hourglass modules to 1 for demonstration. We chose threshold $\theta_d = 1.5$ pixels when matching building vertices with reference vertices, and K was set to 150 and 300 for the custom and the Roof3D dataset respectively. In LoI pooling, we selected $n_1 = 32$ points and $n_2 = 8$ after max-pooling.

In the second stage of 3-D polygonization, threshold of prediction scores θ_s was set empirically to 0.95. We selected 20 sample points in RANSAC plane fitting to reduce the impact from noise, and set the maximal number of iterations to 500.

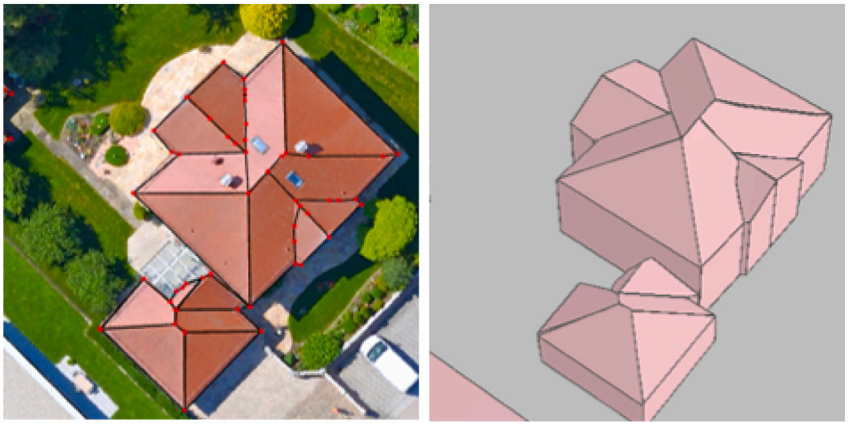
Table 1 shows our model variants. We present the results based on multi-class building representation as base design, with the rest of architecture identical to the baseline model L-CNN (Zhou et al., 2019). "nDSM-backbone" model uses our designed network with separate RGB

[1] https://www.nframes.com
[2] https://www.ldbv.bayern.de/produkte/3dprodukte/gelaende.html



(a) Raw predictions, polylines (b) Closed roof lines, polylines (c) Final roof planes, polygons

Fig. 5. Example of our proposed roof plane closing algorithm. Dangle points exist in the unprocessed polylines, which makes polygon generation impossible. After leveraging information from raster predictions, complete roof plane polygons are reconstructed.



(a) 2-D roof plane polygons (b) LoD-2.2 building model

Fig. 6. Example of a LoD-2.2 building model. The 3-D model is built by extruding 2-D roof plane polygons, in (a), to 3-D roof polygons with added vertical walls and projected ground, in (b).

and nDSM pre-encoding, but without skip connection for the two sets of pre-encoded features. "Skip-connect" is with skip connection of pre-encoded features switched on in backbone network. "nDSM-vectorizer" is the model where we concatenate nDSM with the input features for connectivity learning. "Buffered-LoI" is the model with our more robust LoI pooling module.

4.3. Evaluation

4.3.1. Evaluation of raster predictions

We evaluate the predicted building parts map using F1 score, as semantic evaluation. F1 score is calculated as

$$\mathrm{F1} = \frac{2A_{t\cap\hat{t}}}{A_t + A_{\hat{t}}}, \tag{3}$$

where A_t and $A_{\hat{t}}$ are the areas of reference and prediction respectively, $A_{t\cap\hat{t}}$ is the area of the intersection of reference and prediction.

Intersection over union (IoU) is also widely used as a metric to evaluate similarity of two objects, defined as

$$\mathrm{IoU} = \frac{A_{t\cap\hat{t}}}{A_{t\cup\hat{t}}} \times 100\%, \tag{4}$$

where $A_{t\cup\hat{t}}$ is the area of the union of reference and prediction.

We use IoU to evaluate reconstructed polygons.

Table 1
Ablation study models. We divide our experiments into backbone-related and vectorizer-related, concerning different network modules. "X" indicates switched on for a specific design. See main text for details of each model variant.

Backbone related			
Name	Multi-class	Include nDSM	Skip connection
L-CNN			
Multi-class	X		
nDSM-backbone	X	X	
Skip-connect	X	X	X
Vectorizer related			
Name	Backbone related	Include nDSM	Buffered LoI pooling
nDSM-vectorizer	X	X	
Buffered-LoI	X	X	X

4.3.2. Evaluation of vector predictions

We adapt average precision (AP) from object detection and evaluate our predicted building vertices, building roof lines and roof planes. AP is calculated based on precision and recall, defined as

$$\mathrm{precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{5}$$

$$\mathrm{recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{6}$$

where TP, FP and FN are numbers of samples of true positive, false positive and false negative respectively.

Fig. 7. Overview of study area and distribution of training (red rectangles) and testing (green rectangles) areas. Training and testing areas were selected randomly. Annotations aim to cover all possible types and shapes of buildings in the study area. After tiling, we collected 1,379 training samples and 732 testing samples. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We adapt mean AP (Zhou et al., 2019) for building vertices and roof lines. For roof plane polygons which do not have prediction scores, we report mean precision (mP) and mean recall (mR) to evaluate the quality of predicted polygons. The term "mean" refers to averaging across different IoU thresholds when counting TP, FP and FN in sense of polygons.

Additionally, we report mean IoU (mIoU) for each pair of prediction and reference. Each prediction is matched with the reference polygon with the maximal IoU value if this IoU value exceeds a threshold, otherwise we consider this prediction as unmatched. Mean IoU is defined as the averaged IoU values of all matched prediction–reference pairs. We report mIoU without penalty as the averaged IoU of all matched pairs, and mIoU with penalty as the averaged IoU of all matched and unmatched pairs, where unmatched IoU values are set to 0%.

To evaluate locations of building vertices, we calculate polygon difference (PD) as

$$\mathrm{PD}\left(P_1, P_2\right) = \frac{1}{N\left(P_1\right)} \sum_{\mathbf{p}\in P_1}^{P_1} \min \|\mathbf{p}, P_2\|, \tag{7}$$

where $N\left(P_1\right)$ is the number of vertices of polygon P_1, $\mathbf{p}$ is the coordinate of each vertex in polygon P_1, and $\|\mathbf{p}, P_2\|$ is the Euclidean distance of vertex $\mathbf{p}$ to each vertex in polygon P_2.

We report bidirectional PDs as $\mathrm{PD}_{t,p} = \mathrm{PD}\left(P, \hat{P}\right)$ and $\mathrm{PD}_{p,t} = \mathrm{PD}\left(\hat{P}, P\right)$ to measure the polygonal differences based on vertex locations, where P and $\hat{P}$ are reference and predicted polygons respectively.

In contrast to location performance, complexity of reconstructed polygons is evaluated based on ratio of vertex numbers (RN), calculated as

$$\mathrm{RN} = \frac{N\left(\hat{P}\right)}{N\left(P\right)}. \tag{8}$$

We consider PD and RN only for matched prediction–reference pairs. The final results are the averaged PD and RN over all matched prediction–reference pairs in a dataset.

Table 2
Quantitative evaluation results. The switched on modules can be found in Table 1. F1 scores from 1 to 3 are building outlines, roof lines and roof parts respectively. APv and APl are AP for vertices and lines respectively. Other than mAPv and mAPl, we also report APv at threshold 1.5 pixels and APl at threshold 3 pixels. The best results are highlighted in bold text. Units are percentage.

Name	$F1_{p,1}$	$F1_{p,2}$	$F1_{p,3}$	mAPv	APv@1.5	mAPl	APl@3
L-CNN	–	–	–	50.8	54.8	37.8	34.2
Multi-class	53.9	45.3	85.3	56.2	60.7	44.3	40.2
nDSM-backbone	**58.3**	51.8	**88.0**	62.6	67.0	51.5	47.6
Skip-connect	56.7	**53.1**	87.5	63.9	68.3	54.2	49.8
nDSM-vector	57.2	50.7	87.7	63.2	68.0	**54.5**	**50.4**
Buffered-LoI	57.2	52.5	87.7	**64.3**	**69.1**	54.0	50.0

5. Results and discussion

5.1. Evaluation on custom dataset

5.1.1. Evaluating raw outputs

We present evaluation results based on different variants of our models to evaluate validity of each module. For raster evaluation, we

Table 3
Evaluation results of predicted polygons. We report mP, mR and mIoU that are averaged over different IoU thresholds. Two specific IoU thresholds are presented, i.e. 50% and 75%. Specially for mIoU, we report the evaluation results for with penalty (before /) and without penalty (after /). Bidirectional PDs and RN are calculated for each prediction–reference pair matched at IoU threshold 50%. PDs are in unit of pixels while the other units are percentage. The best results are highlighted in bold text.

Name	mP	P@50	P@75	mR	R@50	R@75
nDSM-vector	46.4	59.0	50.6	**54.7**	**68.9**	**59.4**
Buffered-LoI	**52.2**	**66.7**	**56.1**	47.7	60.6	51.1
Name	mIoU	IoU@50	IoU@75	$PD_{t,p}$	$PD_{p,t}$	RN
nDSM-vector	41.7/90.8	51.2/**86.7**	45.8/90.5	**4.4**	**9.3**	**1.2**
Buffered-LoI	**47.1/91.0**	**57.8**/86.6	**51.0/91.0**	4.6	**9.3**	**1.2**

report F1 scores for building outlines, roof lines and roof parts. For vector evaluation, we report mean AP for building vertices (mAPv) and building roof lines (mAPl). We choose from 0.5 to 3 pixels with step 0.5 (including 3) as distance thresholds for mAPv, and from 3 to 9 pixels with step 2 for mAPl.

As is shown in Table 2, improvements for all metrics from baseline (L-CNN) to our designed multi-class building representation model are observed. By leveraging both RGB and nDSM, the performance is further improved significantly, with increase in mAPv and mAPl of 6.4% and 7.2% respectively.

With our proposed skip connection, we were able to further improve the quality of predictions, with mAPv increased from 62.6% to 63.9% and mAPl from 51.5% to 54.2%. Interestingly, a small decrease concerning raster predictions is observed, especially for building outlines and roof parts. It can be explained by the fact that the pre-encoded features from the RGB and nDSM branches are not refined enough as to improve the raster predictions, rather harm the performance. However, the separated semantic (RGB) and geometric (nDSM) features are able to improve the vector related performance.

As expected, using nDSM in connectivity learning is beneficial, which further increased mAPl from 54.2% to 54.5%, but surprisingly with small decrease in mAPv. Furthermore, although "Buffered-LoI" achieved the best performance regarding building vertices, but mAPl is slightly smaller than the non-buffered LoI pooling model.

The observations above suggest that the performances of the extraction of building vertices and of the reconstruction of roof lines might be a trade-off. We suspect that it is because of the information flow in LoI pooling. The LoI pooling samples points along line that include non-corner edge points, and the corresponding features are used to learn the connectivity between extracted points. In this process, the pooled features from each sampled point are encouraged to be more homogeneous, since they all indicate positive connection, and this information is back-propagated to the extraction of building vertices. However, for the task of building vertices extraction, the edge point features and the corner point features should be more distinguishable. Consequently, a trade-off appears between the task of building vertices extraction and the task of connectivity learning. Therefore, better performance regarding building vertices does not necessarily lead to better performance in sense of building roof lines, and vice versa.

5.1.2. Evaluating 2-D roof plane polygons

We evaluate the quality of reconstructed 2-D roof plane polygons and present the evaluation results in Table 3. We used IoU thresholds from 50% to 95% with interval 5% to calculate mIoU, and present IoU values at threshold 50% and 75%. Prediction–reference pairs were matched at IoU level 50% for PDs and RN.

The two model variants reported in Table 3 achieved matching results regarding building vertices and roof lines (shown in Table 2), but differences are more distinguishable in polygon evaluation.

First of all, it is observed that the non-buffered LoI pooling model ("nDSM-vector") performs better in favor of recall while the buffered LoI model ("Buffered-LoI") achieved higher precision. It suggests that with buffered LoI pooling, more predicted polygons are correct but with less portion of successfully extracted reference polygons.

This conclusion is also supported by the results regarding IoU. Higher IoU values are observed for all IoU related metrics for "Buffered-LoI", with the exception of IoU without penalty at threshold 50% being 0.1% lower than the non-buffered LoI model. It further confirms that with our proposed buffered LoI pooling module, roof plane polygons of higher quality are reconstructed.

The non-buffered LoI pooling model, on the other hand, tends to be more exhaustive in extracting reference polygons at the cost of producing more poor quality polygons, i.e. higher recall but lower precision and IoU. This observation indicates that the proposed buffered LoI pooling helps solve the problem of misalignment of reference annotations and input images: when there are a large amount of shifted annotations, it is expected for the model to output more shifted polygons, resulting in more wrong polygons (lower precision) but larger portion of correctly retrieved reference polygons (higher recall). If this shift is corrected by the network, precision is expected to increase and recall is expected to drop. Therefore, our proposed buffered LoI pooling should perform better dealing with datasets of more poorer annotations in sense of annotation shifts.

Very high mIoU for without penalty is observed for both models, suggesting that both models are able to generate polygons close to manual delineation. However, low mIoU with penalty suggests that our method missed reference polygons. Typical failure cases are shown in Fig. 8. Such scenarios are difficult to handle for the methods which are optical photogrammetry based. When the texture difference of two objects is not distinguishable enough, our method can hardly extract the corresponding polygons.

Our method achieved good performance in sense of localizing building vertices. In average, our predicted vertices are around 0.5 m away from reference vertices. Furthermore, complexity of reconstructed polygons is satisfying, with RN close to 1 for both models.

5.1.3. Qualitative evaluation

Fig. 9 shows a collection of different types of buildings and the corresponding reconstructed polygons. Overall, our method is able to produce accurate polygons with sharp corners. Finer details such as small dormer windows are successfully reconstructed. Complicated scenes, e.g. as is shown in Fig. 9(b), are handled satisfyingly.

There are still some issues to be solved. Firstly, it seems that shadow leads to wrong polygons in some cases, as is highlighted in Fig. 9(d). This is surprising, since the shadow issue with RGB images is expected to be solved by utilizing nDSM. One possible explanation is that in some cases, the information from RGB images is somewhat more dominant than the information from nDSM, resulting in residual impacts of shadows, especially when the pre-encoded RGB features are directly used in connectivity learning. Secondly, background roof-top objects could result in redundant vertices in reconstructed polygons and introduce small distortions.

For 3-D building reconstruction, we processed the whole city of Landsberg am Lech in Germany for qualitative evaluation. Fig. 10 shows four examples of LoD-2.2 building models with increasing building complexity. With our proposed method, smaller dormers are correctly extracted in most cases. Complicated buildings, shown in Fig. 10(c) and 10(d), are also reconstructed with satisfying correctness, especially on finer details.

The reconstructed models are not perfect. Fig. 11 shows two typical failure cases. Due to false negative predictions of building roof lines, incomplete roof objects are obtained (Fig. 11(a)) and merged roof planes lead to wrong reconstruction (Fig. 11(b)).

To conclude, based on quantitative and qualitative results, our proposed method is able to reconstruct high quality LoD-2.2 building models, in spite of large variations of building structures and complexities. False negative predictions, especially missing roof lines, can result in incomplete or wrong building models.

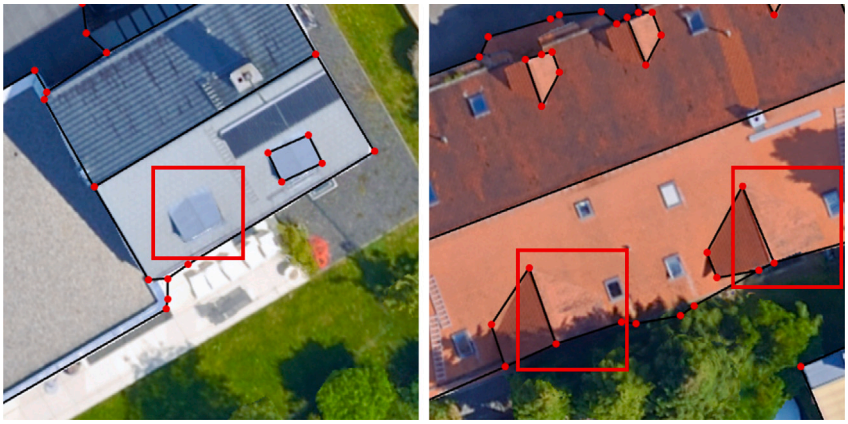


Fig. 8. Examples of typical false negative samples (marked with red rectangles). These two example areas are visualized as RGB images overlaid with reconstructed polygons. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 9. Examples for qualitative evaluation. Areas of interest are highlighted with red rectangles. Generally speaking, our method produces accurate and visually pleasing polygons, even in very complicated scenes. However, it seems that shadow influences our model in some cases, resulting in wrongly reconstructed polygons. Other roof-top objects, e.g. chimneys, produces redundant vertices. More examples can be found in appendix. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.2. Evaluation on Roof3D

For the Roof3D dataset, we compare our method with Plane4LoD2 (Schuegraf et al., 2024), as is shown in Table 4.

Overall, mAPv and mAPl are both much smaller than the results using the custom dataset. This is explained by the fact that the training data provided in the Roof3D dataset is of poor quality, especially the references from public sources, as is shown in Fig. 12. Since connectivity learning plays an important role in our proposed method, high quality and accurate annotations are required, i.e. correct locations and connectivity of building vertices. When false information is given, wrong connections are captured, resulting in errors being propagated throughout the network.

The methods proposed by Schuegraf et al. (2024) are all raster based. It is reported that the result was improved using channel and

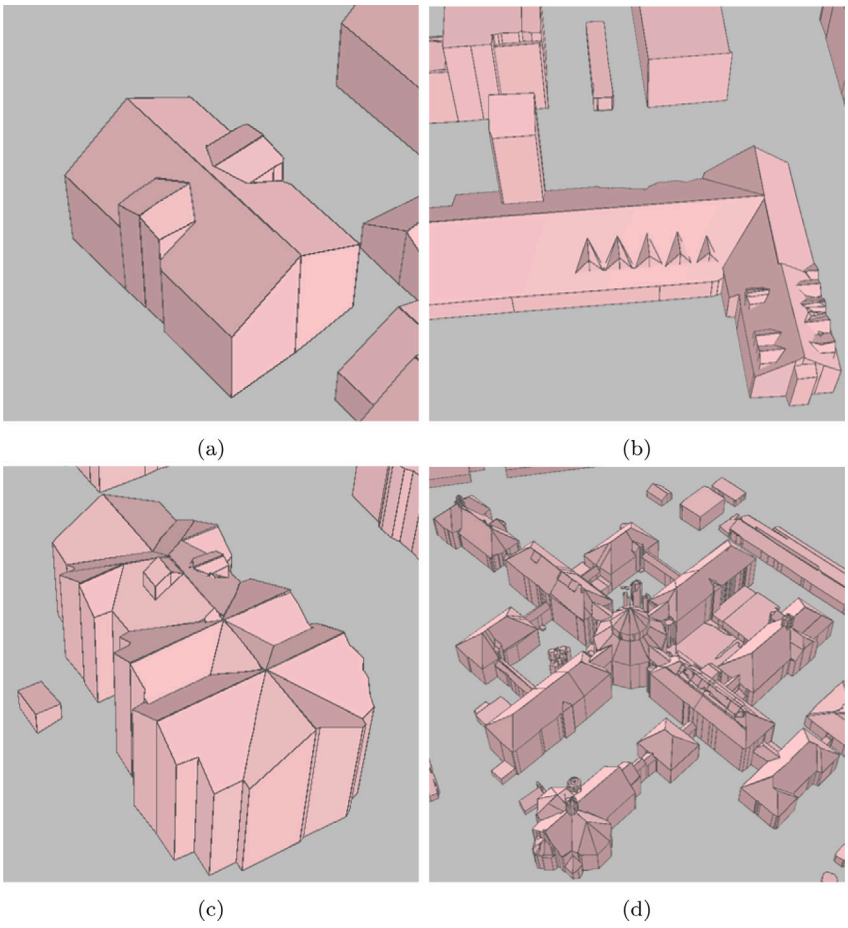


Fig. 10. Examples for qualitative evaluation of LoD-2.2 models. Different buildings with varying complexity are visualized. Overall, the proposed method is able to produce high quality polygonal LoD-2.2 building models with fine details. It is also capable of handling varying types and structures of buildings. More examples can be found in appendix.

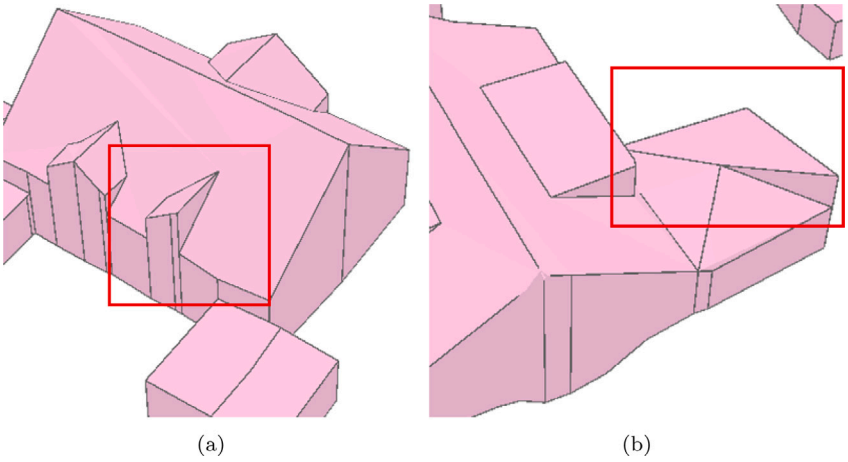


Fig. 11. Examples of failed reconstructions. Typical failure cases are missing roof objects and merged roof planes.

spatial attention, which encourages the network to focus more on specific channels and spatial locations. We believe that this mechanism helps the network identify wrongly annotated areas, in which the attention gets higher and "forces" the model to predict incorrect results. Therefore, the impact of false annotations is less dominant, resulting in higher mAPp than our proposed method.

The influence of incorrect references, especially shifts between annotations and images, is further confirmed by mAPl of the two models with non-buffered and buffered LoI pooling. While in Table 2, "nDSM-vector" and "Buffered-LoI" have very close mAPl (54.5% and 54.0% respectively), but for Roof3D, the difference is much larger: the buffered LoI pooling model is 6.1% higher than the non-buffered model. It proves that the buffered LoI pooling design improves model performance when references of poorer quality have to be used.

Additionally, the GSD of the images in Roof3D is 0.3 m. Since the raster predictions of our proposed method in its current form are of size fourth of the input images, real building vertices could be aggregated into one pixel, leading to information loss. However, this issue could be solved by replacing the backbone network with a network that outputs

Fig. 12. Examples of two annotations in the Roof3D training data. False reference data exist in large amount, leading to drastic performance drop.

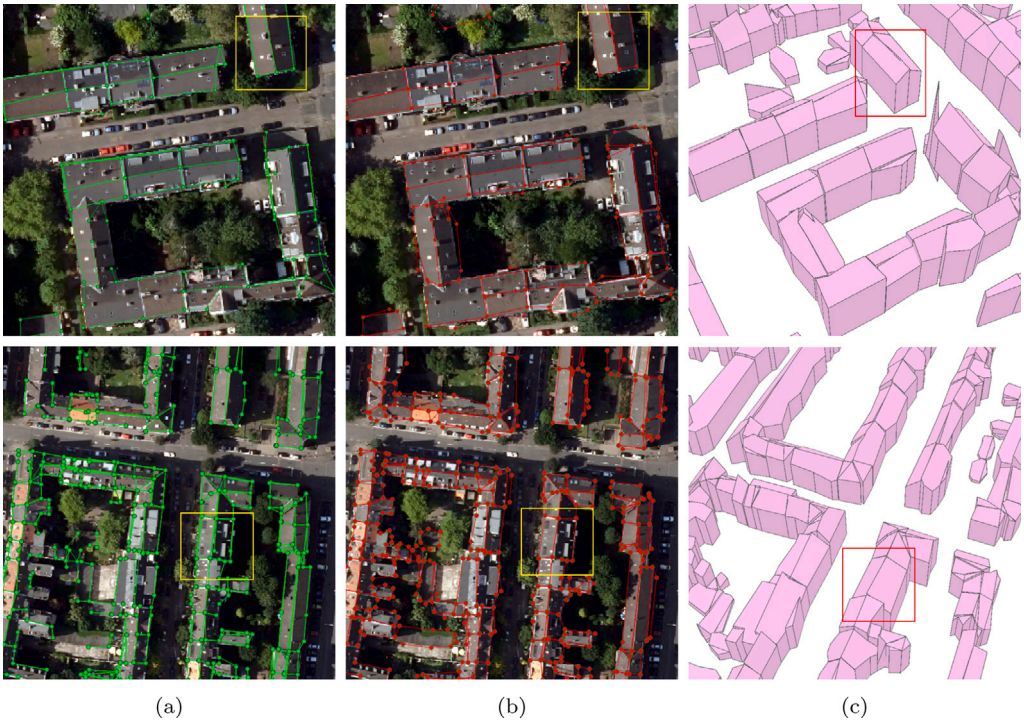


Fig. 13. Examples of results from Roof3D. Each column from left to right shows predicted building roof lines, reconstructed 2-D polygons and final LoD-2 building models. Areas that show fairly good performances are highlighted with colored rectangles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
Evaluation results of the Roof3D dataset. "mAPp" denotes mean AP of polygons. Note that all prediction scores were set to 1 for mAPp calculation. For Plane4LoD2, we present three different backbones, and refer to Schuegraf et al. (2024) for details.

Name	mAPv	mAPl	mAPp
Fuse-UResNet34	–	–	11.9
Plane4LoD2-UResNet34	–	–	12.7
Plane4LoD2-EfficientUnetB3	–	–	13.8
nDSM-vector	52.9	19.2	12.4
Buffered-LoI	52.5	25.3	11.7

predictions of the same spatial dimensions as the inputs. We include this point into our future outlooks.

Fig. 13 shows two examples of predictions using our proposed method. When comparing with the results based on the custom dataset, more false positive and false negative predictions exist, leading to poorer quality of reconstructed building models, e.g. zigzag boundaries, broken polygons for complete roof planes. Moreover, height incontinuity at polygon borders appears more frequently, suggesting that the 3-D polygonization stage requires higher quality of DSMs to obtain better building models. Nevertheless, when the roof lines are well extracted, the corresponding generated polygons and 3-D building models have good quality, with two example areas highlighted using colored rectangles in Fig. 13.

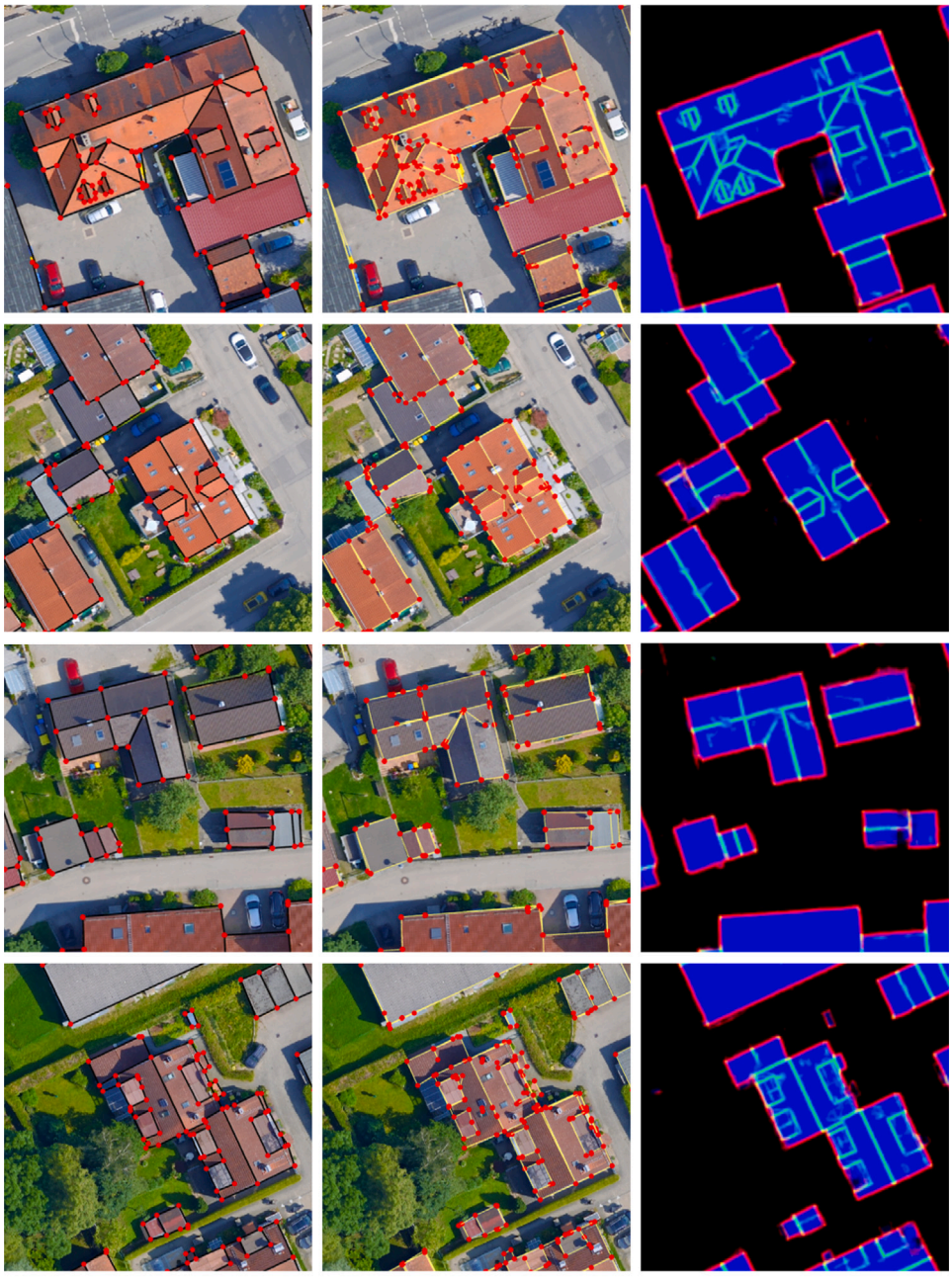


Fig. A. Additional examples for qualitative evaluation from the custom dataset. From left to right each column presents: (a) predicted polygons with RGB overlay, (b) predicted polylines with RGB overlay, and (c) raster prediction. Best viewed zoomed-in electronically.

6. Conclusion

In this paper, we present a novel neural network to extract 2-D building roof planes based on photogrammetry products and demonstrate a method to generate 3-D polygonal building models. Experiments show that our method is able to automatically reconstruct high quality and satisfying 3-D building models, for both LoD-2 and LoD-2.2.

Our proposed backbone network utilizes RGB and nDSM to extract semantic building models and achieved high accuracy in sense of raster predictions. Furthermore, the extracted building roof lines have quality close to manual delineation, as is suggested by the evaluation metrics and visual inspection. In the end, we are able to fill the gap between extraction of roof lines and obtaining roof polygons and provide high quality polygons of LoD-2.2 building models with finer details, which are useful in many important decision making processes.

However, limitations still exist in our work. Firstly, as a method that is essentially based on RGB images, there are some cases (usually for dormers) where the extraction and polygonization of roof objects are hardly possible. Secondly, in its current form, the performance of the proposed model on datasets with larger GSD and poorer annotations is compromised, which requires further study. Thirdly, topology of roof objects and associated roof planes is only hinted in this work. Explicitly modeled topological relations could better the final LoD-2.2 models. Future works will be focused on solving mentioned issues.

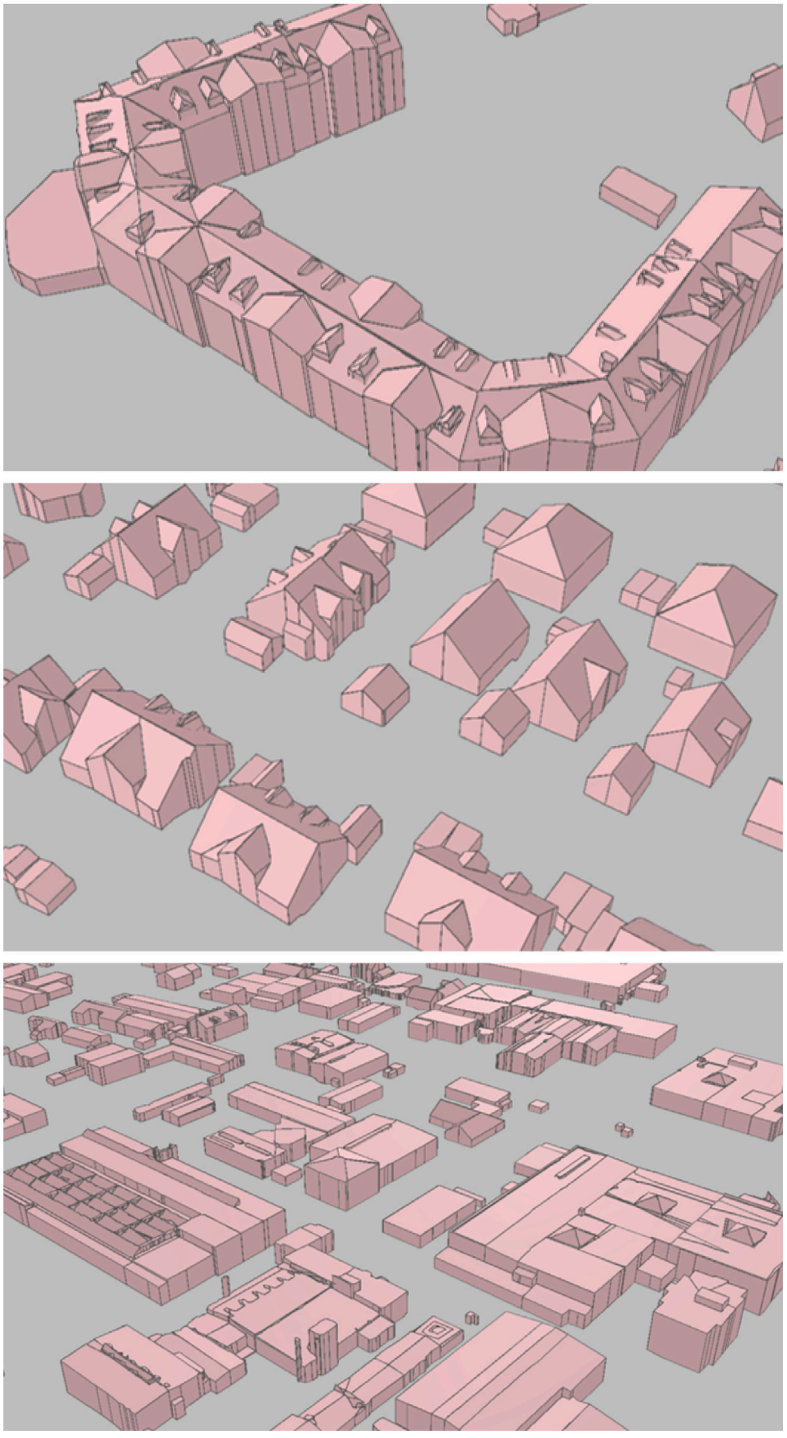


Fig. B. Additional examples for qualitative evaluation on LoD-2.2 building models and 3-D scenes from the custom dataset. Best viewed zoomed-in electronically.

CRediT authorship contribution statement

Yajin Xu: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation. **Juilson Jubanski:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Investigation, Formal analysis, Data curation, Conceptualization. **Ksenia Bittner:** Writing – review & editing, Validation, Supervision, Methodology, Investigation. **Florian Siegert:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

This work was supported by Federal Ministry for Digital and Transport in Germany (BMDV) with grant number 19F2193B, and Bavarian Ministry of Economic Affairs, Regional Development and Energy with grant number DIK0309/01.

Appendix

See Figs. A and B.

References

Arefi, H., Reinartz, P., 2013. Building reconstruction using dsm and orthorectified images. Remote Sens. 5, 1681–1703. http://dx.doi.org/10.3390/rs5041681.

Biljecki, F., Ledoux, H., Stoter, J., 2016. An improved lod specification for 3d building models. Comput. Environ. Urban Syst. 59, 25–37. http://dx.doi.org/10.1016/j.compenvurbsys.2016.04.005.

Bittner, K., Adam, F., Cui, S., Körner, M., Reinartz, P., 2018. Building footprint extraction from vhr remote sensing images combined with normalized dsms using fused fully convolutional networks. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 11, 2615–2629. http://dx.doi.org/10.1109/JSTARS.2018.2849363.

Brenner, C., 2005. Building reconstruction from images and laser scanning. Int. J. Appl. Earth Obs. Geoinf. 6, 187–198. http://dx.doi.org/10.1016/j.jag.2004.10.006, Data Quality in Earth Observation Techniques.

Chen, S., Shi, W., Zhou, M., Zhang, M., Xuan, Z., 2022. Cgsanet: A contour-guided and local structure-aware encoder–decoder network for accurate building extraction from very high-resolution remote sensing imagery. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 15, 1526–1542. http://dx.doi.org/10.1109/JSTARS.2021.3139017.

Douglas, D., Peucker, T., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: Int. J. Geographic Inf. Geovisualization 10, 112–122. http://dx.doi.org/10.3138/FM57-6770-U75U-7727.

Fischler, M., Bolles, R., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24, 381–395. http://dx.doi.org/10.1145/358669.358692.

Hensel, S., Goebbels, S., Kada, M., 2021. Building roof vectorization with ppgnet. Int. Archives Photogramm., Remote Sens. Spatial Inf. Sci. XLVI-4/W4-2021 85–90. http://dx.doi.org/10.5194/isprs-archives-XLVI-4-W4-2021-85-2021.

Huang, W., Tang, H., Xu, P., 2022. Oec-rnn: Object-oriented delineation of rooftops with edges and corners using the recurrent neural network from the aerial images. IEEE Trans. Geosci. Remote Sens. 60, 1–12. http://dx.doi.org/10.1109/TGRS.2021.3076098.

Huang, K., Wang, Y., Zhou, Z., Ding, T., Gao, S., Ma, Y., 2018. Learning to parse wireframes in images of man-made environments. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 626–635. http://dx.doi.org/10.1109/CVPR.2018.00072.

Klasing, K., Althoff, D., Wollherr, D., Buss, M., 2009. Comparison of surface normal estimation methods for range sensing applications. In: 2009 IEEE International Conference on Robotics and Automation. pp. 3206–3211. http://dx.doi.org/10.1109/ROBOT.2009.5152493.

Li, Z., Wegner, J., Lucchi, A., 2019. Topological map extraction from overhead images. In: 2019 IEEE/CVF International Conference on Computer Vision. ICCV, pp. 1715–1724. http://dx.doi.org/10.1109/ICCV.2019.00180.

Liu, J., Chen, H., Yang, S., 2021. Research on building dsm fusion method based on adaptive spline and target characteristic guidance. Information 12, http://dx.doi.org/10.3390/info12110467.

Mahmud, J., Price, T., Bapat, A., Frahm, J., 2020. Boundary-aware 3d building reconstruction from a single overhead image. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 438–448. http://dx.doi.org/10.1109/CVPR42600.2020.00052.

Mousa, Y., Helmholz, P., Belton, D., Bulatov, D., 2019. Building detection and regularisation using dsm and imagery information. Photogramm. Rec. 34, 85–107. http://dx.doi.org/10.1111/phor.12275.

Newell, A., Yang, K., Deng, J., 2016. Stacked hourglass networks for human pose estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.), Computer Vision – ECCV 2016. Springer International Publishing, Cham, pp. 483–499.

Qian, Z., Chen, M., Zhong, T., Zhang, F., Zhu, R., Zhang, Z., Zhang, K., Sun, Z., Lü, G., 2022. Deep roof refiner: A detail-oriented deep learning network for refined delineation of roof structure lines using satellite imagery. Int. J. Appl. Earth Obs. Geoinf. 107, 102680. http://dx.doi.org/10.1016/j.jag.2022.102680.

Robinson, C., Ortiz, A., Park, H., Gracia, N., Kaw, J., Sederholm, T., Dodhia, R., Ferres, J., 2022. Fast building segmentation from satellite imagery and few local labels. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. CVPRW, IEEE Computer Society, Los Alamitos, CA, USA, pp. 1462–1470. http://dx.doi.org/10.1109/CVPRW56347.2022.00152.

Schuegraf, P., Fuentes Reyes, M., Xu, Y., Bittner, K., 2023. Roof3d: A real and synthetic data collection for individual building roof plane and building sections detection. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. pp. 1–9.

Schuegraf, P., Shan, J., Bittner, K., 2024. Planes4lod2: Reconstruction of lod-2 building models using a depth attention-based fully convolutional neural network. ISPRS J. Photogramm. Remote Sens. 211, 425–437. http://dx.doi.org/10.1016/j.isprsjprs.2024.04.015.

Zhao, K., Kang, J., Jung, J., Sohn, G., 2018. Building extraction from satellite images using mask r-cnn with building boundary regularization. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. CVPRW, pp. 242–2424. http://dx.doi.org/10.1109/CVPRW.2018.00045.

Zhao, W., Persello, C., Stein, A., 2022. Extracting planar roof structures from very high resolution images using graph neural networks. ISPRS J. Photogramm. Remote Sens. 187, 34–45. http://dx.doi.org/10.1016/j.isprsjprs.2022.02.022.

Zhou, Y., Qi, H., Ma, Y., 2019. End-to-end wireframe parsing. In: 2019 IEEE/CVF International Conference on Computer Vision. ICCV, pp. 962–971. http://dx.doi.org/10.1109/ICCV.2019.00105.

Zorzi, S., Bazrafkan, S., Habenschuss, S., Fraundorfer, F., 2022. Polyworld: Polygonal building extraction with graph neural networks in satellite images. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 1938–1947. http://dx.doi.org/10.1109/CVPR52688.2022.00189.

Zorzi, S., Bittner, K., Fraundorfer, F., 2021. Machine-learned regularization and polygonization of building segmentation masks. In: 2020 25th International Conference on Pattern Recognition. ICPR, pp. 3098–3105. http://dx.doi.org/10.1109/ICPR48806.2021.9412866.