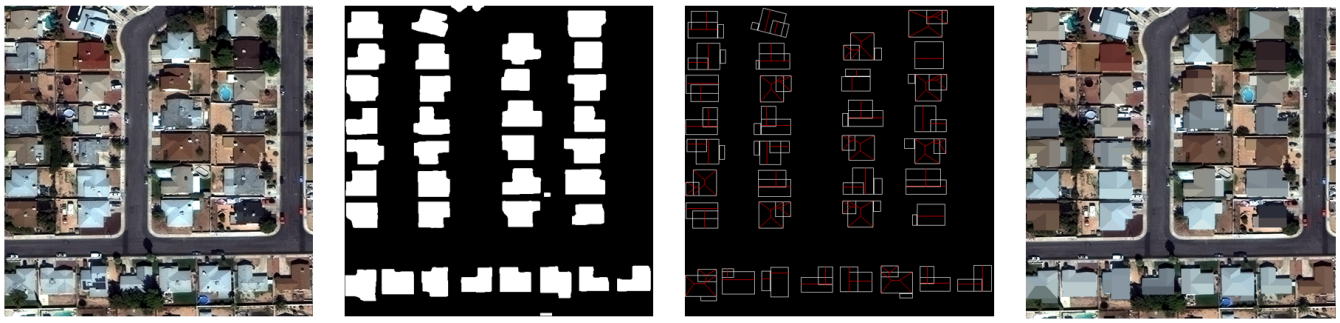
DOI: 10.1111/cgf.14472
EUROGRAPHICS 2022 / R. Chaine and M. H. Kim
(Guest Editors)
Volume 41 (2022), Number 2

Procedural Roof Generation From a Single Satellite Image

Xiaowei Zhang and Daniel Aliaga

Purdue University, West Lafayette, IN, USA



(a) Input urban Area (b) Initial segmentation (c) Our decomposed roofs and roof ridges (d) Our procedural output

Figure 1: *Our method automatically generates procedural roofs for an input urban area. (a) An input urban area from SpaceNet dataset [VLB18]. (b) The initial segmentation (Mask R-CNN [HGDG17]) of (a). (c) Our decomposed roof parts and predicted ridges. (d) Our generated procedural roofs rendered on top of real image (a).*

Abstract

Urban procedural modeling has benefited from recent advances in deep learning and computer graphics. However, few, if any, approaches have automatically produced procedural building roof models from a single overhead satellite image. Large-scale roof modeling is important for a variety of applications in urban content creation and in urban planning (e.g., solar panel planning, heating/cooling/rainfall modeling). While the allure of modeling only from satellite images is clear, unfortunately structures obtained from the satellite images are often in low-resolution, noisy and heavily occluded, thus getting a clean and complete view of urban structures is difficult. In this paper, we present a framework that exploits the inherent structure present in man-made buildings and roofs by explicitly identifying the compact space of potential building shapes and roof structures. Then, we utilize this relatively compact space with a two-component solution combining procedural modeling and deep learning. Specifically, we use a ***building decomposition component*** *to separate the building into roof parts and predict regularized building footprints in a procedural format, and use a* ***roof ridge detection component*** *to refine the individual roof parts by estimating the procedural roof ridge parameters. Our qualitative and quantitative assessments over multiple satellite datasets show that our method outperforms various state-of-the-art methods.*

CCS Concepts

• ***Computing methodologies*** → ***Shape analysis; Image-based rendering;***

1. Introduction

Urban procedural modeling has had great success in computer graphics due to its ability to generate detailed 3D content for a variety of applications including animation and games, city planning, autonomous driving, and urban sustainability. Procedural modeling methods exploit man-made patterns and their regularity (in our case of urban structures: walls are straight and parallel, corners have predetermined angles, etc.) in order to succinctly express the possible shapes. By having the ability to quickly and automatically model real world urban areas, and to easily edit them, procedural modeling enables many what-if scenario tools as well as flexible content creation.

Since the seminal paper of Parish and Müller [PM01], numerous works have concentrated on city-scale modeling [VABW09; AFS*11], road modeling [CEW*08; GPMG10], parcel modeling [VKW*12], building modeling [MWH*06; WWSR03], and facade

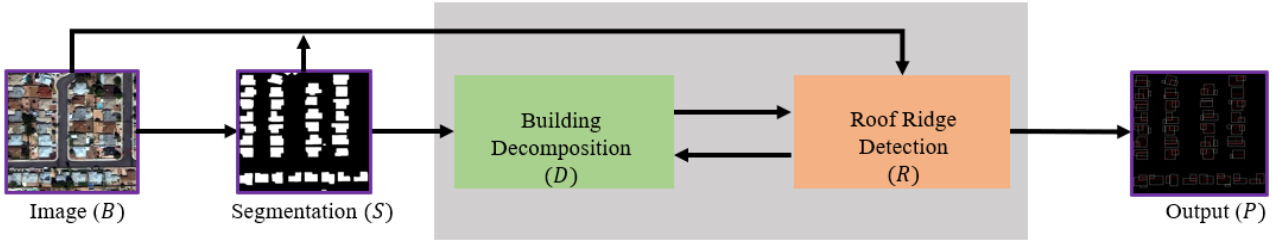


Figure 2: Pipeline. *Our approach consists of a building decomposition component and a roof ridge detection component. A single satellite image gets segmented by a building instance segmentation model (e.g., Mask R-CNN [HGDG17]). And our components directly work on the initial segmentation image and generate procedural urban output.*

modeling [BSW13; MZWG07; SHFH11; ZXJ*13]. Since it is difficult to define and cumbersome to write detailed procedural models of large areas, many works have focused on automatic creation of procedural models, or inverse procedural modeling, often starting from one or more photographs; for example, city-scale modeling [VGA*12; KFWM17; ZSBA20], tree creation [SPK*14; HBDP17], building modeling [NGA*16; NBA18; ZWF18], and facade generation [ZMA20]. However, few works have focused on automatically generating procedural roofs from images and almost none from a single image. In this paper, we address automatic generation of procedural roofs using only a single satellite image. Hence, the approach is highly scalable and can be used to create content representing roofs from many cities worldwide, thereby enabling large-scale content creation, supporting urban planning applications such as for solar panels, heating/cooling design, and more.

Our key inspiration is that while a single image does not contain significant z-values (i.e., 3D) information, we can exploit that the space of possible man-made building roofs in typical urban settings is constrained. In particular, we identify that the space of potential building shapes and roofs (PBSR) can be explicitly enumerated, enabling $> 90\%$ coverage as observed in our preliminary experiments. Hence, together with the aforementioned regular properties of man-made structures, we can robustly infer urban geometrical details from a single satellite image, despite noise and occlusions. With the help of a deep learning framework, we demonstrate creating accurate procedural models of roofs exceeding the ability of prior methods.

Our automatic procedural modeling approach consists of two main components. First as a preprocess, we take advantage of recent deep segmentation models (e.g., Mask R-CNN [HGDG17]) to produce an initial building instance segmentation of the input area (Figure 1 (b)). Then, our *building decomposition component* partitions the building image according to the PBSR space (i.e., a deep learning framework exploiting the potential building shapes and roofs space). This results in a regularized building footprint (e.g., straight walls, parallel walls, corners with predetermined angles, symmetrical arrangements) represented as a parameterized procedural building footprint and a set of initial procedural roof parts (Figure 1 (c)). Further, our *roof ridge detection component* refines the procedural roof structure by inferring ridge parameter values of individual roof parts (Figure 1 (c)). The final output is a synthetic procedural generation of the initial input area, as shown in Figure 1 (d).

Our framework yields both improved results over prior methods and produces discrete vector-based procedural structures, all from a single satellite image without the need of high-resolution aerial images, LiDAR or point-cloud data. In our comparisons to multiple techniques used well-established datasets, our method is consistently better than prior work both quantitatively and qualitatively. As far as we know, our work is *the first pipeline to handle building footprint regularization, building decomposition, and roof ridge prediction all together given a single un-annotated satellite image*. We anticipate our work will inspire various future directions for urban structure modeling at a large scale.

Our approach is summarized in Figure 2. As a preprocessing step, individual building images $\{b_1, ...b_i, ...\}$ and corresponding segmentations $\{s_1, ...s_i, ...\}$ are extracted from the input satellite image B and the segmentation image S (e.g., s_i is cropped from S based on a loose oriented bounding box of the building instance). These are then given to the building decomposition D and roof ridge detection R components to yield the procedural output P. In summary, our main contributions are as follows:

- a novel potential building shape and roof space (PBSR) which is relatively compact and able to express over $> 90\%$ of the roofs found in the well-known datasets used (Section 3.1),
- a framework to generate regularized building footprints (Section 3.2) and roof parts (Section 3.3) in a parameterized procedural representation, and
- a synthetic training dataset which provides a flexible combination of building shapes and roof types suitable for various deep learning networks and avoids time-consuming and expensive data annotations.

2. Related Work

Our work builds on procedural and inverse procedural modeling, generative modeling (e.g., deep learning), and building footprint extraction and roof reconstruction in order to automatically produce procedural roofs from a single satellite image.

2.1. Procedural and Inverse Procedural Modeling

One popular method that offers an effective way of generating complex, parameterized 3D urban models is procedural and inverse procedural modeling [PM01; MWH*06; WWSR03; VAB10; VGA*12; BYMW13; RMGH15; DAB16; NGA*16; KFWM17; NBA18; ZSBA20]. Procedural programs generate high-quality and human-editable urban geometry when executed. Inverse procedural modeling finds a procedural representations of provided input data. Demir et al. [DAB16] used similarities in architecturals models to inversely generate procedural models. Kelly et al. [KFWM17] described a method to fuse street-level imagery, GIS footprints, and a coarse 3D mesh to produce 3D urban building mass and facade models. Nishida et al. [NBA18] present an interactive tool that allows users to automatically generate a procedural building from a single image of the building. Zeng et al. [ZWF18] trains deep neural networks to procedurally apply shape grammar rules and reconstruct CAD-quality models from 3D points. Nevertheless, none of these methods focus on roof reconstruction from a single (overhead) satellite image. More recently, Zhang et al. [ZSBA20] introduced an automatic approach to generate a 3D urban procedural model, based on a segmented and labeled satellite image. However, this work focuses on low-resolution satellite imagery ($\geq 3m$ per pixel) and aims to generate statistically and visually similar. Besides, it requires additional population, elevation, and Open Street Maps datasets.

2.2. Generative Modeling

In contrast, the recent explosion of research in deep learning has led to deep generative models [RMC16; KGS*18; KALL18; MKKY18; XZH*18; KLA19; NCC*20; QZF20] which can be applied to 3D modeling. Given enough training data, theoretically they can learn to generate plausible urban structures with broad variability. In particular, Kelly et al. [KGS*18] introduce a pipeline to automatically and realistically decorate building mass models by adding semantically consistent geometric details and textures. House-GAN [NCC*20] employs a generative adversarial network for floor-plan generation, while requiring room adjacency relations as input. Subsequently, Roof-GAN [QZF20] presents a novel generative adversarial network that generates structured geometry of residential roof structures as a set of roof primitives and their relationships. However, these approaches are aiming to generate plausible urban structures with broad variability, and not focusing on accurate reconstruction. Their outputs often yield unrealistic outcomes and representations that are challenging to further edit, especially when considering intricate structural details and structural regularities of urban spaces. Besides, the aforementioned methods typically depend on a set of well-annotated datasets to train deep neural models, and none of them works on a single satellite image.

2.3. Building Footprint Extraction

Many state-of-the-art deep segmentation networks (e.g., [LSD15; RPB15; CPK*15; BKC17; CPK*17; ZDS*18; CZP*18; TAJF19]) can be applied to building footprint extraction. Specifically, Fully Convolutional Networks (FCNs) [LSD15] introduce deconvolution via upsampling operations and provide an alternative to fully connected layers in classification models. U-Net [RPB15] infers high-resolution feature maps by joining the top-down and bottom-up pathways with lateral connections. DeepLab [CPK*15; CPK*17; CZP*18] maintains high-resolution by replacing strided convolution with atrous convolution. However, these approaches include many more content pixels than boundary pixels. This imbalance causes them to produce inaccurate building/roof edges.

To solve this challenge, polygon-based building boundary delineation work has been proposed. These methods focus mainly on active contours and on edge (point) assembling. In [MTK*18; CLFU19], they present frameworks which utilize the strengths of both CNNs and active contour models [KWT88] to produce an end-to-end polygon-based output model. Although the active contour approaches improve mask coverage compared to the aforementioned CNN-based semantic segmentation, blob-like contours that do not match building boundaries are produced. In [LWL19; NF20; ZNF20], these approaches start with detecting/extracting building primitives (e.g., corners, edges or regions) using CNNs and then employ other techniques (e.g., RNNs, integer programming or Graph Neural Networks) to assemble them leading to polygon-based building outputs. However, the usability of these methods is limited because they cannot predict complex shapes because of deficiencies in primitive feature extraction and detection modules. Moreover, the RNN and GNN modules are computationally expensive and [NF20; ZNF20] require extra building corner and edge annotations. Recently, Li et al. [LLM20] design an algorithm pipeline ASIP to improve the work of [LWL19] by extracting and vectorizing objects in images with polygons – we compare to this method, amongst others, in our results section.

2.4. Building and Roof Reconstruction

Building and roof reconstruction is an active research area. However, it usually requires multiple data sources (e.g., LiDAR, DSM, DTM, point clouds, etc.) and few works focus on reconstruction from a single satellite image. Arefi and Reinartz [AR13] directly detect roof ridges utilizing high resolution DSMs and orthorectified satellite images. Zheng and Zheng [ZZ17] propose a hybrid approach, combining the data- and model-driven approaches to generate LoD2 building models by using LiDAR, 2D building footprints and high resolution orthophoto images. Li et al. [LYT*20] present a novel approach to segment the roof planes from airborne LiDAR point clouds using hierarchical clustering and boundary relabeling. Ywata et al. [YDSdO21] introduce a method to extract building roof boundaries in object space by integrating a high-resolution aerial images stereo pair and three-dimensional roof models reconstructed from LiDAR data. In [AAT19; AAH20], they work on a deep learning-based approach to detect and reconstruct roof parts of buildings from a single image. However, they require high resolution *aerial images* and annotate the dataset for roof ridges and building boundaries. In [MPBF20; WZB21], they present deep learning based approaches for automatic 3D building reconstruction. However, they need elevation data (e.g., DSM) for training and their results are not regularized. None of the mentioned approaches automatically reconstruct roofs using only a single satellite image as input.

3. Procedural Generation

In this section, we first give a general overview of the potential building shapes and roofs space, and then describe the building decomposition component including the processing details, synthetic data creation and training of neural models. Then, we present the roof ridge detection component in a similar manner. Note: Having building decomposition component and roof ridge detection component separate can significantly reduce the required training data (i.e., if not, each building part must support diverse roof types with different roof ridge configurations) and makes training more efficient.

Table 1: *Variables and their meanings.*

variable	meaning
B	an input satellite image of an urban area
b_i	a building image of B
S	the segmented image of B
s_i	the segmentation of b_i
m_i	the edge map of b_i
D	building decomposition component
t_i^D	the building shape family of b_i
c_i^D	the building configuration of b_i
r_{ij}	the j_{th} rectangle in c_i^D
R	roof ridge detection component
e_{ij}	the edge map of r_{ij}
t_{ij}^R	the roof shape family of r_{ij}
c_{ij}^R	the ridge configuration of r_{ij}

To assist with terminology, we provide a table summarizing the subsequent variables (Table 1).

3.1. Potential Building Shapes and Roofs (PBSR)

The PBSR is an approximation to represent all possible building and roof structure combinations. For this purpose, we propose a graph representation: each node stands for a singular roof and each edge signifies the connection of adjacent roof parts.

3.1.1. Building Shape Families.

Considering the number of nodes (or roof parts see Figure 3) and the different connection scenarios of roof parts, the number of possible building shape families is:

$$\sum_i F_i \qquad and \qquad F_i = \sum_j T_{ij} \tag{1}$$

where F_i is the i_{th} building shape family defined according to the number of nodes. T_{ij} is the j_{th} topology of F_i, and T_{ij} is defined by the connections within F_i. For example, the building shape families of F_1, F_2, F_3 and F_4 in Figure 4 consist of one, two, three and four roof parts respectively. By means of graph isomorphism and the assumption of the graph being connected, F_1 and F_2 can only have one topology, F_3 can have two topologies, and F_4 can have up to four topologies. Intuitively, the building shape "L" and "T" belong to T_{21}. T_{32} includes "U" and "Z" building shapes (see Figure 3). A closed four-side building shape is in T_{43}. However, T_{31}, T_{41}, T_{42} and T_{44} are possible shapes but not common in the real world. In summary, we can theoretically grow the space of possible building shape families to larger values for i and account for any actual building. However, as discussed later, by limiting the possible set of building shape families, and their configurations, to a rather compact number, we can practically capture most buildings in our datasets. For example, considering i up to 4 results in 8 possible parameterized building shape families.

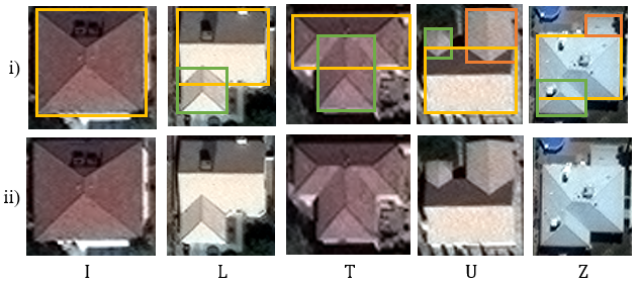


Figure 3: Roof parts. *We show roof parts of certain building shapes (I, L, T, U, and Z). For each, i) one or more roof parts in different colors. ii) the corresponding building image.*

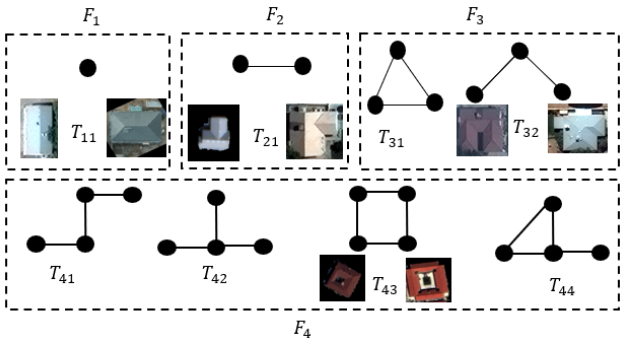


Figure 4: Potential building shape families. *We show building shape families of F_1, F_2, F_3 and F_4. See main text for more details.*

3.1.2. Roof Families.

As for roof families, we consider two edge types appearing in typical roofs: external edges (e.g., eaves) and internal edges (e.g., ridges and hips). We assume the perimeter of a single roof part are the external edges and the internal ridges follow the main direction of the roof (e.g., parallel or perpendicular to eaves). Hips are the internal edges that connect ridges to corners. In our current implementation, we support flat, gable, hip, pyramid and half-hip roof families (see Figure 5) which includes most common types according to [ZW15; PHK*15; PFA*17].

3.2. Building Decomposition Component

In this part, we describe the processing details of our building decomposition component D, and also how we create a synthetic dataset and train the classifier. As shown in Figure 6, the component D consists of three parts: the previous described PBSR, a building

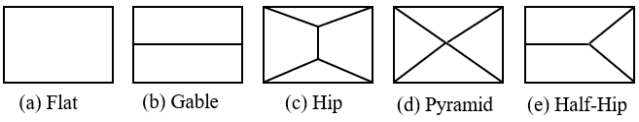


Figure 5: Potential Roof Families. *We show our supported roof families.*

shape family classifier, and a building family configuration recognizer.

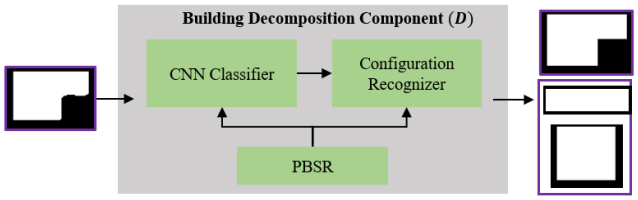


Figure 6: *Building decomposition component.*

3.2.1. Classifier

Taking the segmentation image s_i as input, the CNN-based classifier predicts the building shape family t_i^D. The classifier network is a ResNet [HZRS15] with a modification of the last fully-connected layer to having the number of supported building shape families. We train the classifier with 185,700 synthetic images (see Section 3.2.3) and achieve 96.5% classification accuracy when testing on segmentation images s_i.

3.2.2. Configuration Recognizer

Next, we need to recognize the precise configuration of the determined building shape family. This enables producing a specific parameterized procedural output for the building footprint and subsequently for estimating roof parameters.

The configurations are determined by the arrangements of parameter values of roof parts. The parameters for each single roof part (or node) is $r = \{x, y, w, h\}$ (assuming it's a rectangular roof) where (x, y) is its top-left corner and (w, h) is its size. Thus, for all building shape families, the total number of possible configurations is

$$\sum_i \sum_j \sum_k C_{ijk} \quad \text{and} \quad C_{ijk} = \{r_{ijk1}, r_{ijk2}, ..., r_{ijkn}\} \tag{2}$$

where C_{ijk} is the k_{th} configuration of T_{ij} and contains $n (n \geq 1)$ roof parts. However, this exhaustive list has redundancies that we seek to omit to achieve better performance (e.g., faster search). We ignore configurations that are affine transformations of other configurations (e.g., translation, flip, mirror, etc.). Further, we split the image into grids (setting grid size to 2 pixels) and iterate parameters in the grid space. Additionally, according to our preeliminary analysis of our used portion of SpaceNet [VLB18], roughly 90% of building shapes are covered by I, L, T, U, and Z (see Figure 3) . Hence, we only consider the families T_{11}, T_{21}, T_{32} and T_{43}. In summary, we support 4520 configurations in total.

To search for the configuration of the current building shape family t_i^D, we apply the following strategies. Since the configurations only consists of canonical instances without those generated by transformations, we apply a series of image processing methods to s_i; e.g., cropping the s_i, resizing the s_i to size $(120, 120)$, centering the s_i with 4 pixels margin, and then applying transformations including flipping s_i horizontally or vertically, and rotating s_i (e.g., 90, 180, 270 degrees) before searching for the configuration. Eventually, we find the best match $c_i^D = \{r_{i1}, ..., r_{ij}, ...r_{iq_i}\}$ using intersection-of-union (IOU) (the best IOU of transformed s_i and the generated footprint image based on a configuration). The matched c_i^D will be passed to the roof ridge detection component for subsequent processing.

3.2.3. Synthetic Data Creation and Training

While assuming building image inputs (e.g., b_i) is an option, it requires to manually annotate a large set of real-world training images for both building footprints and roof ridges. Instead, we leverage synthetic dataset to avoid labor-intensive annotations and thus can do self-supervised training more easily.

Regarding the creation of a synthetic dataset, we consider building structure regularities. Buildings exhibit properties such as straight walls, parallel walls, walls meeting at one of a set of predetermined angles (e.g., 90 or 135 degrees), symmetrical arrangements, and other features. For the sake of simplicity, we focus on straight walls, parallel walls and right angle regularities meaning a rectangle represents each single roof part. Our synthetic dataset consists of all the configurations of building shape families discussed previously. Additional types can be added to our dataset easily.

Moreover, in order to handle noisy and irregular building footprint segmentation, aside from typical data augmentation techniques (e.g., flip, translation, rotation, etc.), we add noise to our clean/regularized synthetic images (see Figure 7 (a)). We apply random occlusion or bumps (e.g., different shapes and sizes) around the footprint boundaries. Based on preliminary experiments, we furthermore include different levels of transformations (e.g., low-noisy, medium-noisy and high-noisy: see Figure 7 ii) from left to right) when training the classifier.

Figure 7: Data Transformation. *We show (a) building footprints, and (b) roofs. For each, i) clean and regularized synthetic images. ii) Images after corresponding transformations.*

3.3. Roof Ridge Detection Component

In the following, we provide details about estimating procedural roof parameters, creating synthetic roof dataset, and training the

roof family classifier. The component R (see Figure 8) also has three parts: the previously used PBSR, a roof type classifier, and a roof ridge configuration recognizer.

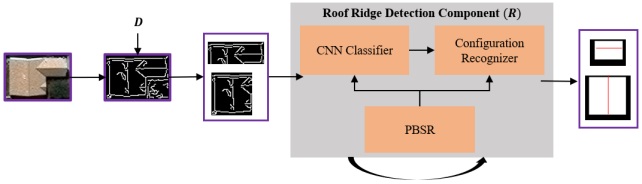


Figure 8: *Roof ridge detection component.*

3.3.1. Classifier

Given a building image b_i, we first apply histogram equalization to adjust color contrast, and then make use of an edge detector ([Can86] as default) to generate an initial edge map m_i. m_i is subsequently cropped into an edge set $\{e_{i1},...e_{ij},...e_{iq_i}\}$ following the aforementioned configuration c_i^D of b_i. For any e_{ij}, the classifier predicts the roof family type for the corresponding roof part (or node). The classifier network is a ResNet [HZRS15] with modification of the last fully-connected layer to the number of supported roof family types. We train the classifier with 119,500 synthetic images and achieve 95.6% classification accuracy when testing on edge maps e_{ij}.

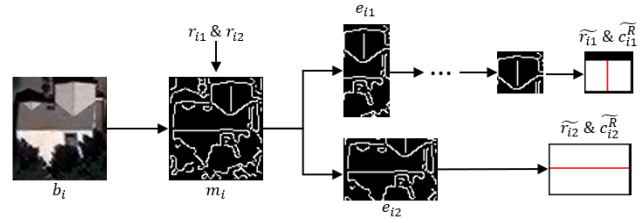


Figure 9: Roof Processing Step. *We show an example to illustrate how our recognize roof ridges and refine roof parts.*

3.3.2. Configuration Recognizer

Given an edge set $\{e_{i1},...e_{ij},...e_{iq_i}\}$ of b_i, the goal is to detect and estimate the roof structures for the whole building. For example, if $q_i = 1$, b_i consists of a single roof. We recognize the roof family type and find the best matched configuration c_{i1}^R by maximizing supporting points in e_{i1} and the candidate ridge coverage. The supporting points are those edge points of e_{i1} whose distance to the candidate ridge line is smaller than a threshold value (e.g., setting to 5% of the length of the candidate). If $q_i \geq 2$, we need to refine the sizes of roof parts (see roof part r_{i1} in Figure 9). We start with recognizing the roof family type for each e_{ij}. Based on the result, we decide the main roof part (e_{i2} in Figure 9). Afterwards, we focus on iteratively refining the rest of the roof parts. We start with decreasing the overlap area by half each time (binary search by following the direction perpendicular to the main roof). For each iteration, we predict the roof family type and and find the corresponding best matched configuration until we find the best matching (e.g., setting candidate ridge coverage to 90% and maximizing the number of supporting points). In the end, it leads to the final roof set $\tilde{c^D}_i = \{\tilde{r}_{i1},..\tilde{r}_{ij},...\tilde{r}_{iq_i}\}$ with their corresponding ridge configuration set $\{\tilde{c^R}_{i1},..\tilde{c^R}_{ij},...\tilde{c^R}_{iq_i}\}$.

Finally, we collect the roof set and roof ridge configurations of each b_i, and combine them into procedural output for B.

3.3.3. Synthetic Data Creation and Training

We generate synthetic roof images to support the roof family types in Section 3.1. For the purpose of representing noisy and irregular edge maps e_{ij}, aside from typical data augmentation techniques, we further transform the synthetic roof images by adding random noisy curve lines and randomly removing small parts of the edges (see Figure 7 (b)). During training, similarly we apply different levels of transformations.

4. Implementation And Results

Our method is implemented in Python and we train our neural network models using PyTorch. The weights of our classifiers are trained by the SGD optimizer where initial learning rate is set to 1e-3. Our typical input image sizes are $(H,W,C) = (128,128,1)$. It runs on an Intel i9 workstation with NVIDIA RTX 2080 8GB cards. We quantitatively and qualitatively evaluate our approach on multiple satellite datasets.

4.1. Datasets

We test our components on three satellite datasets across different regions in the world and at different spatial resolutions (30 cm and 50 cm).

SpaceNet: This dataset [VLB18] contains building footprints in five cities across the world. In our experiments, we use the Las Vegas region (because it has small off-nadir angle which minimizes foreshortening effects) which contains over 3,800 tiles of 200m x 200m areas with a spatial resolution of 30 cm. Each tile comes with an 650 x 650 pixel RGB satellite image, a high-resolution panchromatic image, a low-resolution multi-spectral image, and ground truth building footprint annotation. For our purposes, we only use the RGB satellite image, and we train a footprint segmentation model based on Mask R-CNN [HGDG17] as the benchmark.

CrowdAI: The CrowdAI dataset[MCK*20] contains 340,000 total tiles with 300 by 300 pixel RGB images at a 30 cm spatial resolution. Building footprint annotations are also provided. Regarding the segmentation model, we directly use the provided Mask R-CNN to generate the initial segmentation results.

Urban3D: Urban3D dataset contains 236 tiles of 2048 x 2048 pixel images and annotations with a spatial resolution of 50 cm. Each RGB tile in this dataset is accompanied by its Depth Surface Model and Digital Terrain Model (DSM and DTM), which provides high-resolution building height information. We train a segmentation model based on DeepLabv3+ [CZP*18] to achieve the initial segmentation.

4.2. Evaluation Metrics

We rigorously evaluate our outputs from both building decomposition component and roof ridge detection component. In particular, we evaluate both pixel-wise correctness and structure regularization for building footprint. For roof ridges, we assess the performance in terms of a hit ratio, completeness, and correctness metric.

4.2.1. Building Footprint

For pixel-wise correctness evaluation, we use the following statistical measures:

$$\begin{aligned} Accuracy &= \frac{TP+TN}{ALL} \\ Precision &= \frac{TP}{TP+FP} \\ Recall &= \frac{TP}{TP+FN} \\ F1 &= 2 * \frac{Precision * Recall}{Precision + Recall} \end{aligned} \tag{3}$$

with true positives TP, false positives FP, true negatives TN, and false negatives FN for building and non-building.

For building footprint regularization evaluation, we follow the observation that building walls are typically parallel or meet at corners of predetermined angles (90, 45 or 135 degrees). Hence, for the polygonal outline of a building footprint, we compute the interior angles in degrees (within $[0,180]$) for each vertex. Then, we cluster corners of similar angles into a group g_i and all groups form part of the set G. The regularization error of E_r is defined as:

$$E_r = \sum_{g_i} \frac{stdvar(g_i)}{scale(g_i)} + w_r||G||, \tag{4}$$

where $stdvar(g_i)$ measures the standard deviation of angles in g_i, $scale(g_i)$ is used to approximately normalize the error – we set $scale(g_i)$ to 5 in our experiments. $||G||$ is the number of corner groups. We add $||G||$ to encourage fewer and thus larger groups. w_r is a weight that balances the two aforementioned terms – we set $w_r = 0.1$ in our tests. It's easy to recognize that a rectangular building footprint whose walls are parallel and corners are all 90 degrees has $E_r = 0.1$ since the $stdvar(g_i) = 0$ and $||G|| = 1$.

4.2.2. Roof Ridges

With regard to roof ridge evaluation, we adapt and modify the relevant definitions of correctness and completeness from [HMWJ97]. Correctness represents the percentage of the predicted roof ridge which lies within a rectangular buffer around the ground truth ridge. We set the buffer width to be 0.1 times the length of ground truth roof ridge (which results in a width of typically 2-4 pixels). Using a similar strategy, completeness is the percentage of the ground truth which lies within the buffer around the predicted ridge. Hence, we define correctness and completeness for roof ridges as follows:

$$Correctness = \frac{length\ of\ matched\ prediction}{length\ of\ prediction}, \tag{5}$$

$$Completeness = \frac{length\ of\ matched\ ground\ truth}{length\ of\ ground\ truth}, \tag{6}$$

In order to be consistent with footprint evaluation, we also evaluate ridges per entire building. Since a single building commonly contains more than one roof ridge, we apply weights to balance the importance of the multiple ridges based on their length. For easy illustration, we assume the building has a set of ridge lines $\{l_1,...,l_i,....\}$ and the corresponding weight set is $\{w_1,...,w_i,....\}$. We define:

$$w_i = \begin{cases} \frac{||l_i||}{\sum_j ||l_j||} & \text{if } i_{th} \text{ ridge is found (predicted)} \\ 0 & \text{otherwise} \end{cases}$$

where $||l_i||$ is the length of the ridge l_i. "Found" means that both correctness and completeness of the ridge are bigger than a threshold (setting to 0.3 in our experiments). Hence, the correctness and completeness per building is

$$\begin{aligned} Correctness\ per\ building &= \sum_i w_i * correctness(l_i) \\ Completeness\ per\ building &= \sum_i w_i * completeness(l_i) \end{aligned} \tag{7}$$

Additionally, we define another term to represent how many of the ridges have been "hit" (or found) by our method. This term also considers the relative importance of each ridge, and thus is computed as a percentage by summing the aforementioned weights of the (found) ridges:

$$Hit\ Ratio = \sum_i ||w_i||, \tag{8}$$

4.3. Building Footprint Comparison

Although our approach aims to generate procedural roofs, regularized and parameterized procedural building footprint are produced by our building decomposition component as an intermediate output. We compare this intermediate result to the outputs of other methods. We selected tiles at random but that are at least almost orthorectified. To be specific, we test on 7 random tiles of SpaceNet (which corresponds to 180 buildings), 10 random tiles of CrowdAI (resulting in another 65 buildings), and 1 random tile of Urban3D (producing another 415 buildings). We compare our generated building footprints to the initial segmentations of each dataset (see Section 4.1) and a state-of-the-art building footprint delineation method ASIP [LLM20]. The initial segmentation for each of the datasets is in the first row in each group of Table 2. For ASIP, we set $\beta = 10^{-3}$ and $\lambda = 10^{-5}$ as recommended by their paper and apply their tool to generate results in a polygon format.

As shown in Table 2, our method is slightly less in accuracy and precision, but always achieves better recall (meaning our results are more complete) and F1 score performance (Note: only F1 of CrowdAI is not the best) for all three datasets compared to the best model in terms of footprint correctness (e.g., for SpaceNet, our accuracy and precision is **0.5%** and **1.1%** lower, but our recall and

Table 2: Quantitative Comparison. *We compare our building footprints with the initial footprint segmentations and the ASIP method [LLM20] for SpaceNet, CrowdAI and Urban3D datasets. For footprint correctness, higher is better. For regularization error, lower is better.* ***Note:*** *our E_r of our approach is* 0.1 *which has been explained in Section 4.2.1.*

Dataset	Method	Footprint Correctness				E_r
		Acc.	Pre.	Rec.	F1	
SpaceNet	Mask R-CNN	**89.5%**	**95.1%**	86.9%	90.4%	—
	ASIP	89.0%	94.6%	86.4%	90.0%	1.79
	Ours	89.0%	94.0%	**88.6%**	**90.8%**	**0.1**
CrowdAI	Mask R-CNN	**94.4%**	**92.3%**	89.8%	**90.8%**	—
	ASIP	93.9%	92.0%	88.7%	90.0%	1.52
	Ours	93.1%	88.9%	**91.9%**	90.2%	**0.1**
Urban3D	DeepLabv3+	**86.4%**	**81.0%**	85.3%	81.6%	—
	ASIP	85.8%	80.3%	84.2%	80.7%	1.60
	Ours	85.5%	79.4%	**88.1%**	**81.8%**	**0.1**

F1 score is improved by **1.6%** and **0.4%**). Regarding regularization error E_r defined in Equation 4, we compare to the polygonal output of the ASIP method. For the initial segmentation (e.g., Mask R-CNN or DeepLabv3+) and the corresponding ground truth, the polygonal representations don't exist and simply computing the E_r term would provide a very large error for those methods. Nevertheless, it is obvious to recognize that there is no regularization for the segmentation (Figure 10 (c)), and the ground truth (Figure 10 (b)) is regularized and its E_r is close to 0.1. As clearly observed, our results significantly improve building footprint regularization (e.g., for SpaceNet, the regularization error is reduced by **94.4%**). Further, as illustrated in Figure 10, the outputs of our method are visually appealing as well.

Figure 10: Qualitative Comparison. *(a) Real images. (b) Ground truth footprints. (c) Initial building footprint segmentations. (d) ASIP results. (e) Our results.*

4.4. Roof Ridge Comparison

We compare our procedural roofs to three methods which approximately perform the same task as us – these are the most similar works we could find that operate on a single image, though two of these use aerial images at 6 times higher resolution. Since annotations of roof ridges for our test datasets are not available, we manually create them by using an image annotator tool VIA [DZ19]. We randomly chose 83, 21 and 26 buildings from SpaceNet, CrowdAI, and Urban3D, respectively, and annotated the roof ridges. We compare our predicted roof ridges to the state-of-the-art method Conv-MPN which predicts building edges [ZNF20] (we only evaluate the roof ridges in Conv-MPN for fairness). As shown in Table 3, our method consistently achieves better performance compared to Conv-MPN (e.g., hit ratio improved by **31.8%**, correctness improved by **27.7%**, and completeness improved by **45.4%** for SpaceNet). Yet more, as demonstrated in Figure 11, our results are qualitatively preferable.

In addition, we compare to the methods in [AAT19; AAH20]. These approaches reconstruct roofs from a single aerial image at a 5 cm spatial resolution using the Potsdam dataset provided by [ISP19]. We compare to these methods by first down-sampling the aerial image to 30 cm resolution (– the resolution of our tested satellite images –) and then apply our method. In addition, we also manually annotate roof ridges. In terms of hit ratio, correctness, and completeness defined in Section 4.2.2, we obtain **97.9%**, **92.9%** and **96.8%** respectively.

However, [AAT19; AAH20] use a different correctness and completeness term to evaluate their roof ridge and other urban structures. For [AAT19], it outputs 43.4% completeness and 4% correctness for just roof ridges (their completeness and correctness is higher when you also consider the building footprint pixels). In [AAH20], they improved results to 57.7% completeness and 81.3% correctness for roof ridges (using the same metrics as [AAT19]). At satellite-level resolutions, the correctness and completeness term they provide does not seem suitable. Nonetheless, we did compute the values using their method and obtained 35.5% completeness and 34.3% correctness at 30 cms per pixel, as opposed to their values at 5 cms per pixel. While our terms are lower than [AAH20], our method operates at 6 times lower resolution because we used satellite images. We also show our output for the tested tile in Potsdam in the Supplemental Figure 2.

Table 3: Quantitative Comparison. *We compare our predicted ridges with Conv-MPN method [ZNF20] for our SpaceNet, CrowdAI and Urban3D datasets. For all three metrics terms, higher is better.* ***Note:*** *since Conv-MPN is not trained on Urban3D originally, we only show our performance for this dataset.*

Dataset	Method	Hit Ratio	Correctness	Completeness
SpaceNet	Conv-MPN	63.2%	58.7%	44.8%
	Ours	**95.0%**	**86.4%**	**90.2%**
CrowdAI	Conv-MPN	63.4%	61.6%	58.6%
	Ours	**96.3%**	**90.9%**	**92.9%**
Urban3D	**Ours**	87.9%	73.5%	81.0%

Figure 11: Qualitative Comparison. *(a) Real images. (b) Ground truth ridge annotations. (c) Conv-MPN results. (d) Ours results.* ***Note:*** *The red lines in (c) are considered as roof ridges.*

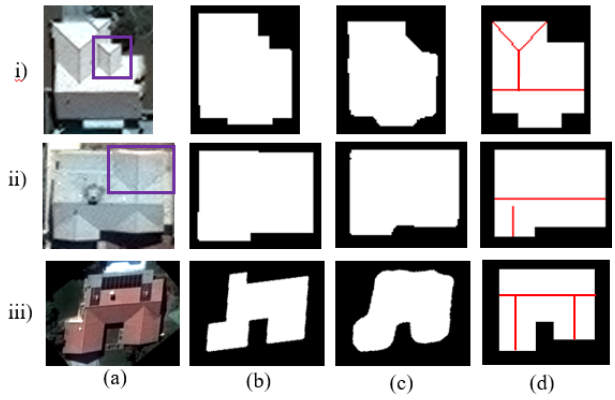


Figure 12: Failure Examples. *(a) Real images. (b) Ground truth footprints. (c) Initial footprint segmentations. (d) Ours results.*

4.5. More Results

We show our procedural urban generations for three large areas in Figure 13. Moreover, since we have a procedural output (instead of an image), we can zoom-in to any part of the area and still have a high-quality result. Additional example are in Supplemental Figure 1.

4.6. Failure Cases

Although we support a very wide range of building and roof types, there are always exceptions (e.g., missing ridges in the edge map, initial segmentation with poor quality, multiple roof parts in one rectangle, etc.). Currently for styles outside our assumptions, our approach gives its best guess. In the example i) of Figure 12, the building type is T_{44} based on Section 3.1 and Figure 4. It doesn't belong to our supported type and our prediction misses one roof part shown in a purple box. As for the example ii), multiple roof parts are contained in one rectangle. Our method assumes a singular rectangle stands for one roof part. Therefore, one roof part in the purple box is undetected. For the last example iii), the building image is not orthorectified. We rotated the building segmentation instance based on a loose oriented bounding box and thus building segmentation is not perfectly aligned (horizontally or vertically). Since our method assumes the regularities of building and roof structures, we generate "over-regularized" results. In theory, our framework can handle these failure scenarios by adding more building and roof types to our synthetic training datasets. This is listed as future work.

5. Conclusion

We propose a novel framework that consists of a building decomposition component and a roof ridge detection component to automatically generate procedural roofs from a single satellite image. Our procedural output for the urban area can be used in many applications. To the best of our knowledge, our work is the first pipeline to handle building footprint regularization, building decomposition, and roof ridge prediction all together given a single un-annotated satellite image. Through comprehensive experiments, we show our approach significantly improves the performance compared to several state-of-the-art methods for multiple datasets. However, our approach has some limitations. Although we support a very wide range of styles, there are always exceptions. Please find examples in Section 4.6.

Our approach has several avenues of future work. For example, since our PBSR is an approximation to represent all possible building and roof structures, we would like to extend our synthetic dataset to support more building shapes and roof types (e.g., non-right angle buildings, not orthorectified satellite image, mansard roof type, etc.). Also, currently we generate the edge map using Canny edge detector which depends on the quality of image and tuning parameters. We would like to explore deep learning based approaches to help with this step. In addition, we would like to detect and reconstruct the details (e.g, dormers, chimneys, etc.) of roofs. Finally, we are also interested in applying our framework to interdisciplinary applications, such as solar panel planning, energy modeling, and more.

References

[AAH20] ALIDOOST, F., AREFI, H., and HAHN, M. "Y-SHAPED CONVOLUTIONAL NEURAL NETWORK FOR 3D ROOF ELEMENTS EXTRACTION TO RECONSTRUCT BUILDING MODELS FROM A SINGLE AERIAL IMAGE". *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* V-2-2020 (2020), 321–328. DOI: 10.5194/isprs-annals-V-2-2020-321-2020.

Figure 13: More examples. *(a) Input urban areas from SpaceNet, CrowdAI and Urban3D respectively. (b) The inital segmentation of (a). (c) Our decomposed roof parts and predicted ridges. (d) Our procedural outputs on top of real image (a).*

URL: https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/V-2-2020/321/2020/ 3, 8.

[AAT19] ALIDOOST, FATEMEH, AREFI, HOSSEIN, and TOMBARI, FEDERICO. "2D Image-To-3D Model: Knowledge-Based 3D Building Reconstruction (3DBR) Using Single Aerial Images and Convolutional Neural Networks (CNNs)". *Remote Sensing* 11.19 (2019). ISSN: 2072-4292. URL: https://www.mdpi.com/2072-4292/11/19/2219 3, 8.

[AFS*11] AGARWAL, SAMEER, FURUKAWA, YASUTAKA, SNAVELY, NOAH, et al. "Building Rome in a Day". *Commun. ACM* 54.10 (Oct. 2011), 105–112. ISSN: 0001-0782. DOI: 10.1145/2001269.2001293. URL: https://doi.org/10.1145/2001269.2001293 1.

[AR13] AREFI, HOSSEIN and REINARTZ, PETER. "Building Reconstruction Using DSM and Orthorectified Images". *Remote Sensing* 5.4 (2013), 1681–1703. ISSN: 2072-4292. URL: https://www.mdpi.com/2072-4292/5/4/1681 3.

[BKC17] BADRINARAYANAN, VIJAY, KENDALL, ALEX, and CIPOLLA, ROBERTO. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Dec. 2017). (Visited on 08/09/2020) 3.

[BSW13] BAO, FAN, SCHWARZ, MICHAEL, and WONKA, PETER. "Procedural facade variations from a single layout". *ACM Trans. Graph.* 32.1 (Feb. 2013), 8:1–8:13. ISSN: 0730-0301. DOI: 10.1145/2421636.2421644. URL: http://doi.acm.org/10.1145/2421636.2421644 2.

[BYMW13] BAO, FAN, YAN, DONG-MING, MITRA, NILOY J., and WONKA, PETER. "Generating and Exploring Good Building Layouts". *ACM Trans. Graph.* 32.4 (July 2013). ISSN: 0730-0301. DOI: 10.1145/2461912.2461977 3.

[Can86] CANNY, J. "A Computational Approach to Edge Detection". *IEEE Transactions on Pattern Analysis & Machine Intelligence* (1986) 6.

[CEW*08] CHEN, GUONING, ESCH, GREGORY, WONKA, PETER, et al. "Interactive Procedural Street Modeling". *ACM Trans. Graph.* 27.3 (Aug. 2008), 1–10. ISSN: 0730-0301. DOI: 10.1145/1360612.1360702. URL: https://doi.org/10.1145/1360612.1360702 1.

[CLFU19] CHENG, DOMINIC, LIAO, RENJIE, FIDLER, SANJA, and URTASUN, RAQUEL. "DARNet: Deep Active Ray Network for Building

Segmentation". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019 3.

[CPK*15] CHEN, LIANG-CHIEH, PAPANDREOU, G., KOKKINOS, I., et al. "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs". *ICLR* (2015) 3.

[CPK*17] CHEN, L., PAPANDREOU, G., KOKKINOS, I., et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". *TPAMI* (2017) 3.

[CZP*18] CHEN, LIANG-CHIEH, ZHU, YUKUN, PAPANDREOU, GEORGE, et al. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation". *ECCV*. 2018 3, 6.

[DAB16] DEMIR, İLKE, ALIAGA, DANIEL G., and BENES, BEDRICH. "Proceduralization for Editing 3D Architectural Models". *2016 Fourth International Conference on 3D Vision (3DV)*. 2016, 194–202. DOI: 10.1109/3DV.2016.28 3.

[DZ19] DUTTA, ABHISHEK and ZISSERMAN, ANDREW. "The VIA Annotation Software for Images, Audio and Video". *Proceedings of the 27th ACM International Conference on Multimedia*. MM '19. Nice, France: ACM, 2019. ISBN: 978-1-4503-6889-6/19/10. DOI: 10.1145/3343031.3350535. URL: https://doi.org/10.1145/3343031.3350535 8.

[GPMG10] GALIN, E., PEYTAVIE, A., MARÉCHAL, N., and GUÉRIN, E. "Procedural Generation of Roads". *Computer Graphics Forum* 29.2 (2010), 429–438. DOI: https://doi.org/10.1111/j.1467-8659.2009.01612.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2009.01612.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2009.01612.x 1.

[HBDP17] HÄDRICH, TORSTEN, BENES, BEDRICH, DEUSSEN, OLIVER, and PIRK, SÖREN. "Interactive Modeling and Authoring of Climbing Plants". *Computer Graphics Forum* 36.2 (2017), 49–61. DOI: https://doi.org/10.1111/cgf.13106. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13106. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13106 2.

[HGDG17] HE, KAIMING, GKIOXARI, GEORGIA, DOLLÁR, PIOTR, and GIRSHICK, ROSS. "Mask R-CNN". *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017 1, 2, 6.

[HMWJ97] HEIPKE, CHRISTIAN, MAYER, HELMUT, WIEDEMANN, C., and JAMET, OLIVIER. "Evaluation of Automatic Road Extraction". *Inter. Arch. Photogramm. Remote Sens.* 32 (Oct. 1997) 7.

[HZRS15] HE, KAIMING, ZHANG, XIANGYU, REN, SHAOQING, and SUN, JIAN. "Deep Residual Learning for Image Recognition". *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385 5, 6.

[ISP19] ISPRS. *2D Semantic Labeling Contest-Potsdam*. 2019. URL: http://www2.isprs.org/commissions/%20comm3/wg4/2d-sem-label-potsdam.html 8.

[KALL18] KARRAS, TERO, AILA, TIMO, LAINE, SAMULI, and LEHTINEN, JAAKKO. "Progressive Growing of GANs for Improved Quality, Stability, and Variation". *ICLR*. 2018 3.

[KFWM17] KELLY, TOM, FEMIANI, JOHN, WONKA, PETER, and MITRA, NILOY J. "BigSUR: Large-Scale Structured Urban Reconstruction". *ACM Trans. Graph.* 36.6 (Nov. 2017). ISSN: 0730-0301. DOI: 10.1145/3130800.3130823. URL: https://doi.org/10.1145/3130800.3130823 2, 3.

[KGS*18] KELLY, TOM, GUERRERO, PAUL, STEED, ANTHONY, et al. "FrankenGAN: Guided Detail Synthesis for Building Mass Models Using Style-Synchonized GANs". *ACM Trans. Graph.* 37.6 (Dec. 2018). ISSN: 0730-0301. DOI: 10.1145/3272127.3275065. URL: https://doi.org/10.1145/3272127.3275065 3.

[KLA19] KARRAS, TERO, LAINE, SAMULI, and AILA, TIMO. "A Style-Based Generator Architecture for Generative Adversarial Networks". *(CVPR)*. June 2019 3.

[KWT88] KASS, MICHAEL, WITKIN, ANDREW, and TERZOPOULOS, DEMETRI. "Snakes: Active contour models". *INTERNATIONAL JOURNAL OF COMPUTER VISION* 1.4 (1988), 321–331 3.

[LLM20] LI, MUXINGZI, LAFARGE, FLORENT, and MARLET, RENAUD. "Approximating shapes in images with low-complexity polygons". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020 3, 7, 8.

[LSD15] LONG, JONATHAN, SHELHAMER, EVAN, and DARRELL, TREVOR. "Fully convolutional networks for semantic segmentation". *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015 3.

[LWL19] LI, ZUOYUE, WEGNER, JAN DIRK, and LUCCHI, AURELIEN. "Topological Map Extraction From Overhead Images". *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019 3.

[LYT*20] LI, LI, YAO, JIAN, TU, JINGMIN, et al. "Roof Plane Segmentation from Airborne LiDAR Data Using Hierarchical Clustering and Boundary Relabeling". *Remote Sensing* 12.9 (2020). ISSN: 2072-4292. URL: https://www.mdpi.com/2072-4292/12/9/1363 3.

[MCK*20] MOHANTY, SHARADA PRASANNA, CZAKON, JAKUB, KACZMAREK, KAMIL A, et al. "Deep Learning for Understanding Satellite Imagery: An Experimental Survey". *Frontiers in Artificial Intelligence* 3 (2020) 6.

[MKKY18] MIYATO, TAKERU, KATAOKA, TOSHIKI, KOYAMA, MASANORI, and YOSHIDA, YUICHI. "Spectral Normalization for Generative Adversarial Networks". *ICLR*. 2018 3.

[MPBF20] MAHMUD, JISAN, PRICE, TRUE, BAPAT, AKASH, and FRAHM, JAN-MICHAEL. "Boundary-Aware 3D Building Reconstruction From a Single Overhead Image". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020 3.

[MTK*18] MARCOS, DIEGO, TUIA, DEVIS, KELLENBERGER, BENJAMIN, et al. English. *Proceedings - 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018*. 31st Meeting of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018 ; Conference date: 18-06-2018 Through 22-06-2018. IEEE computer society, Dec. 2018, 8877–8885. DOI: 10.1109/CVPR.2018.00925 3.

[MWH*06] MÜLLER, PASCAL, WONKA, PETER, HAEGLER, SIMON, et al. "Procedural Modeling of Buildings". *ACM Trans. Graph.* 25.3 (2006) 1, 3.

[MZWG07] MÜLLER, PASCAL, ZENG, GANG, WONKA, PETER, and GOOL, LUC VAN. "Image-based Procedural Modeling of Facades". *ACM Trans. Graph.* 26.3 (July 2007). ISSN: 0730-0301. DOI: 10.1145/1276377.1276484. URL: http://doi.acm.org/10.1145/1276377.1276484 2.

[NBA18] NISHIDA, GEN, BOUSSEAU, ADRIEN, and ALIAGA, DANIEL G. "Procedural Modeling of a Building from a Single Image". *Computer Graphics Forum*. Eurographics 37.2 (2018). URL: https://hal.inria.fr/hal-01810207 2, 3.

[NCC*20] NAUATA, NELSON, CHANG, KAI-HUNG, CHENG, CHIN-YI, et al. "House-GAN: Relational Generative Adversarial Networks for Graph-Constrained House Layout Generation". *ECCV*. 2020 3.

[NF20] NAUATA, NELSON and FURUKAWA, YASUTAKA. "Vectorizing World Buildings: Planar Graph Reconstruction by Primitive Detection and Relationship Inference". *European Conference on Computer Vision*. Springer. 2020, 711–726 3.

[NGA*16] NISHIDA, GEN, GARCIA-DORADO, IGNACIO, ALIAGA, DANIEL G., et al. "Interactive Sketching of Urban Procedural Models". *ACM Trans. Graph.* (2016) 2, 3.

[PFA*17] PARTOVI, TAHMINEH, FRAUNDORFER, FRIEDRICH, AZIMI, SEYEDMAJID, et al. "ROOF TYPE SELECTION BASED ON PATCH-BASED CLASSIFICATION USING DEEP LEARNING FOR HIGH RESOLUTION SATELLITE IMAGERY". *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-1/W1 (May 2017), 653–657. DOI: 10.5194/isprs-archives-XLII-1-W1-653-2017 4.

[PHK*15] PARTOVI, TAHMINEH, HUANG, HAI, KRAUSS, THOMAS, et al. "Statistical building roof reconstruction from worldview-2 stereo imagery". *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XL-3/W2 (Mar. 2015), 161–167. DOI: 10.5194/isprsarchives-XL-3-W2-161-2015 4.

[PM01] PARISH, YOAV I. H. and MÜLLER, PASCAL. "Procedural Modeling of Cities". *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '01. New York, NY, USA: Association for Computing Machinery, 2001, 301–308. ISBN: 158113374X. DOI: 10.1145/383259.383292. URL: https://doi.org/10.1145/383259.383292 1, 3.

[QZF20] QIAN, YIMING, ZHANG, HAO, and FURUKAWA, YASUTAKA. "Roof-GAN: Learning to Generate Roof Geometry and Relations for Residential Houses". *CoRR* abs/2012.09340 (2020). URL: https://arxiv.org/abs/2012.09340 3.

[RMC16] RADFORD, ALEC, METZ, LUKE, and CHINTALA, SOUMITH. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". *ICLR*. 2016 3.

[RMGH15] RITCHIE, DANIEL, MILDENHALL, BEN, GOODMAN, NOAH D., and HANRAHAN, PAT. "Controlling Procedural Modeling Programs with Stochastically-Ordered Sequential Monte Carlo". *ACM Trans. Graph.* 34.4 (2015). ISSN: 0730-0301 3.

[RPB15] RONNEBERGER, O., P.FISCHER, and BROX, T. "U-Net: Convolutional Networks for Biomedical Image Segmentation". *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2015 3.

[SHFH11] SHEN, CHAO-HUI, HUANG, SHI-SHENG, FU, HONGBO, and HU, SHI-MIN. "Adaptive partitioning of urban facades". *ACM Trans. Graph.* 30.6 (Dec. 2011), 184:1–184:10. ISSN: 0730-0301. DOI: 10.1145/2070781.2024218. URL: http://doi.acm.org/10.1145/2070781.2024218 2.

[SPK*14] STAVA, O., PIRK, S., KRATT, J., et al. "Inverse Procedural Modelling of Trees". 33.6 (Sept. 2014), 118–131. ISSN: 0167-7055. DOI: 10.1111/cgf.12282. URL: https://doi.org/10.1111/cgf.12282 2.

[TAJF19] TAKIKAWA, TOWAKI, ACUNA, DAVID, JAMPANI, VARUN, and FIDLER, SANJA. "Gated-SCNN: Gated Shape CNNs for Semantic Segmentation". *ICCV* (2019) 3.

[VAB10] VANEGAS, CARLOS A., ALIAGA, DANIEL G., and BENES, BEDRICH. "Building reconstruction using manhattan-world grammars". *CVPR*. 2010 3.

[VABW09] VANEGAS, CARLOS A., ALIAGA, DANIEL G., BENES, BEDRICH, and WADDELL, PAUL A. "Interactive Design of Urban Spaces Using Geometrical and Behavioral Modeling". *ACM Trans. Graph.* 28.5 (Dec. 2009), 1–10. ISSN: 0730-0301. DOI: 10.1145/1618452.1618457. URL: https://doi.org/10.1145/1618452.1618457 1.

[VGA*12] VANEGAS, CARLOS A., GARCIA-DORADO, IGNACIO, ALIAGA, DANIEL G., et al. "Inverse Design of Urban Procedural Models". *ACM Trans. Graph.* 31.6 (Nov. 2012). ISSN: 0730-0301 2, 3.

[VKW*12] VANEGAS, CARLOS A., KELLY, TOM, WEBER, BASIL, et al. "Procedural Generation of Parcels in Urban Modeling". *Comput. Graph. Forum* 31.2pt3 (May 2012), 681–690. ISSN: 0167-7055. DOI: 10.1111/j.1467-8659.2012.03047.x. URL: https://doi.org/10.1111/j.1467-8659.2012.03047.x 1.

[VLB18] VAN ETTEN, ADAM, LINDENBAUM, DAVE, and BACASTOW, TODD M. "SpaceNet: A Remote Sensing Dataset and Challenge Series". *arXiv e-prints*, arXiv:1807.01232 (July 2018), arXiv:1807.01232. arXiv: 1807.01232 [cs.CV] 1, 5, 6.

[WWSR03] WONKA, PETER, WIMMER, MICHAEL, SILLION, FRANÇOIS, and RIBARSKY, WILLIAM. "Instant Architecture". *ACM Trans. Graph.* 22.3 (July 2003), 669–677. ISSN: 0730-0301. DOI: 10.1145/882262.882324. URL: https://doi.org/10.1145/882262.882324 1, 3.

[WZB21] WANG, YI, ZORZI, STEFANO, and BITTNER, KSENIA. "Machine-Learned 3D Building Vectorization From Satellite Imagery". *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, 1072–1081 3.

[XZH*18] XIA, FEI, ZAMIR, AMIR R., HE, ZHIYANG, et al. "Gibson Env: Real-World Perception for Embodied Agents". *(CVPR)*. June 2018 3.

[YDSdO21] YWATA, MICHELLE S. Y., DAL POZ, ALUIR P., SHIMABUKURO, MILTON H., and de OLIVEIRA, HENRIQUE C. "Snake-Based Model for Automatic Roof Boundary Extraction in the Object Space Integrating a High-Resolution Aerial Images Stereo Pair and 3D Roof Models". *Remote Sensing* 13.8 (2021). ISSN: 2072-4292. URL: https://www.mdpi.com/2072-4292/13/8/1429 3.

[ZDS*18] ZHANG, HANG, DANA, KRISTIN, SHI, JIANPING, et al. "Context Encoding for Semantic Segmentation". *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018 3.

[ZMA20] ZHANG, XIAOWEI, MAY, CHRISTOPHER, and ALIAGA, DANIEL. "Synthesis and Completion of Facades from Satellite Imagery". *Computer Vision – ECCV 2020*. 2020 2.

[ZNF20] ZHANG, FUYANG, NAUATA, NELSON, and FURUKAWA, YASUTAKA. "Conv-MPN: Convolutional Message Passing Neural Network for Structured Outdoor Architecture Reconstruction". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020 3, 8.

[ZSBA20] ZHANG, XIAOWEI, SHEHATA, ALY, BENES, BEDRICH, and ALIAGA, DANIEL. "Automatic Deep Inference of Procedural Cities from Global-Scale Spatial Data". *ACM Trans. Spatial Algorithms Syst.* 7.2 (Oct. 2020). ISSN: 2374-0353 2, 3.

[ZW15] ZHENG, YUANFAN and WENG, QIHAO. "Model-Driven Reconstruction of 3-D Buildings Using LiDAR Data". *IEEE Geoscience and Remote Sensing Letters* 12.7 (2015), 1541–1545. DOI: 10.1109/LGRS.2015.2412535 4.

[ZWF18] ZENG, HUAYI, WU, JIAYE, and FURUKAWA, YASUTAKA. "Neural Procedural Reconstruction for Residential Buildings". *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018 2, 3.

[ZXJ*13] ZHANG, HAO, XU, KAI, JIANG, WEI, et al. "Layered analysis of irregular facades via symmetry maximization". *ACM Trans. Graph.* 32.4 (July 2013), 121:1–121:13. ISSN: 0730-0301. DOI: 10.1145/2461912.2461923. URL: http://doi.acm.org/10.1145/2461912.2461923 2.

[ZZ17] ZHENG, YUANFAN and ZHENG, YAOXING. "A Hybrid Approach for Three-Dimensional Building Reconstruction in Indianapolis from LiDAR Data". *Remote Sensing* 9(4) (Mar. 2017). DOI: 10.3390/rs9040310 3.