Vertex Aided Building Polygonization from Satellite Imagery Applying Deep Learning

1st Yajin Xu
Photogrammetry and Image Analysis
German Aerospace Center
Weßling, Germany
yajin.xu@dlr.de

2nd Philipp Schuegraf
Photogrammetry and Image Analysis
German Aerospace Center
Weßling, Germany
philipp.schuegraf@dlr.de

3rd Ksenia Bittner
Photogrammetry and Image Analysis
German Aerospace Center
Weßling, Germany
ksenia.bittner@dlr.de

2023 Joint Urban Remote Sensing Event (JURSE) | 978-1-6654-9373-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/JURSE57346.2023.10144146

***Abstract*—Building extraction is an important task in many fields. The use of convolutional neural networks has been proven to be of great success in building extraction from satellite images. This paper presents a deep learning based vertex aided building polygonization method, which takes RGB satellite images as input and outputs building polygons. Unlike other methods which rely on vertex extraction followed by polygonization, our method requires neither pre-defined number of vertices nor thresholding to obtain extracted vertices. The proposed method has the advantage of simplicity in sense of model complexity, and achieved good performance with average precision of 48.1% and intersection over union of 84.1%.**

***Keywords*—building vectorization, building extraction, pattern recognition, deep learning**

I. Introduction

Building extraction has been a topic of interest for a long time. Efficient and accurate building extraction methods are needed to support various studies. Deep learning is proven to be powerful in many fields, including building extraction and footprint delineation.

With the development of deep learning and convolutional neural networks (CNNs), the general workflow of building extraction can be defined as first to extract image features using a backbone network [1, 2], and then to delineate building footprints. Usually, the delineation is done by semantic segmentation with post-processing refinement [3, 4] or by edge/corner extraction with additional graph neural network (GNN) models [5–7].

For semantic segmentation based methods, backbone networks are used to predict building segments. The output is then refined to generate the final building prediction. This refinement is treated as a problem of regularization, and carried out through polygon regularization [3], automatic regularization by introducing another CNN [8, 9], or by height filtering based on digital surface model (DSM) [10].

Another approach is to generate a different representation of building footprints, instead of a segmentation mask, as training target. For example, frame field is the representation, in which two orthogonal directions are calculated for each pixel, and it is proven to be successful in building extraction [11]. The authors in [12] convert each pixel into vectors describing 3D cuboids and extract the cuboid with the highest score as the building footprint. Similarly, the authors in [13] represent each roof as 4D vectors with facade facings and this representation is learned by the neural network.

In contrast to representation learning approaches, the strategies which directly extract edges and/or corners from images for building footprint seem more straightforward, since they output directly vectorized data. Instead of generating segmentation maps, the authors in [6] extract classified rooflines and use these lines to reconstruct roof planes. Another approach is to solve the polygonization problem as connecting corner points in a series manner [14, 15]. Extracting both edges and corners has also been studied in [16].

GNNs are helpful in edge/corner based methods. These methods first extract image features using a backbone network, then construct graphs with initial vertices [7] or rooflines [5].

Inspired by aforementioned works, we develop a vertex aided building polygonization algorithm. Our method relies on extraction of building segments and building vertices, and we solve this problem in a regression manner to encounter the problem of imbalance in training data. The workflow is shown in Fig. 1.

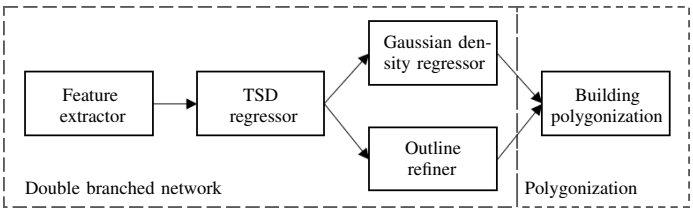


Fig. 1: The workflow of proposed method. Our method first regresses truncated signed distance, then outputs Gaussian density map and segmentation map. These two maps are used for generating building polygons.

Our model consists of two stages. For the first stage, the images are first fed to a feature extractor, and the truncated signed distance (TSD) regressor outputs the predicted TSD map [4] based on the extracted features. The TSD map provides useful information on building boundaries with implication of vertex locations. Two branches are included after the TSD regressor, namely Gaussian density regressor and outline refiner. These two branches predict Gaussian density of building vertex at each location and building boundaries respectively. At the second stage, we perform the vertex aided building vectorization.

The initial polygons are extracted from outline refiner, and are adjusted and regularized by the extracted building vertices.

II. Methodology

Unlike the methods based on GNNs to connect the extracted vertices, we propose an algorithm to guide the connection of the extracted vertices with help of the predicted segmentation map. Our method has the advantage of being simple, and can process the whole scene without multiple stages. The whole architecture of the network is illustrated in Fig. 2.

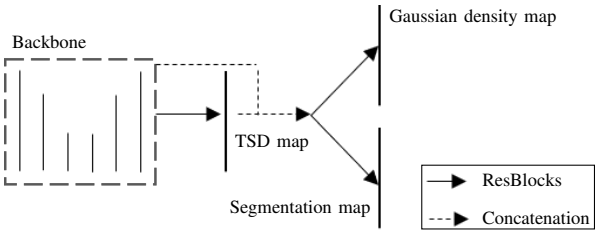


Fig. 2: Double branched network. The backbone is shown on the left with two separate branches on the right.

A. TSD and Gaussian density regression

1) Feature extractor design: We propose a backbone network with U-Net [1] architecture and ResNet-34-like [2] implementation. The down-sizing network extracts features at different scales with skip connections inside each residual block. The up-sizing network takes the concatenated input of upsampled features and previously extracted features in the down-sizing network. This backbone network is shown on the left side in Fig. 2, referred as feature extractor.

2) TSD and Gaussian density: The binary mask as probability map has the advantage in sense of clear semantic interpretation, but learning the probability map directly is detrimental, due to the imbalanced negative and positive training samples. This is true for masks of segments, lines and points. Instead of learning directly the binary masks, we apply the TSD function and Gaussian function to segment and vertex extraction respectively, inspired by [4] and [17].

The TSD function applied in this paper is the same as in [4], defined as

$$TSD\left(\mathbf{p}\right)=\begin{cases}0 & \text{if } \mathbf{p} \text{ along } \mathbf{b},\\ 1+\frac{min(D_{\mathbf{b}}(\mathbf{p}),\tau)}{\tau} & \text{if } \mathbf{p} \text{ inside } \mathbf{b},\\ -1-\frac{min(D_{\mathbf{b}}(\mathbf{p}),\tau)}{\tau} & \text{if } \mathbf{p} \text{ outside } \mathbf{b},\end{cases} \quad (1)$$

where $D_{\mathbf{b}}\left(\mathbf{p}\right)$ is the Euclidean distance between point $\mathbf{p}$ and its nearest point along boundary $\mathbf{b}$, and τ is a hyperparameter which controls the truncation of the signed distance.

The Gaussian density map is obtained by filtering the binary vertex map with a 2-D Gaussian filter, constructed based on 2-D Gaussian function as

$$g\left(\mathbf{p},\mathbf{p_0},\sigma\right)=\frac{1}{2\pi\sigma^2}\exp\left(-\frac{\left(x-x_0\right)^2+\left(y-y_0\right)^2}{2\sigma^2}\right), \quad (2)$$

where $\mathbf{p}_0=(x_0,y_0)$ is the coordinate of peak value, i.e. the coordinate of ground-truth building vertex.

Based on Eq. (2), the density map is then calculated as

$$D=\sum_{i=1}^{N}\max\left(g\left(\mathbf{p},\mathbf{p_i},\sigma\right)\right), \quad (3)$$

where N is the total number of pixels with probability 1, $\mathbf{p}$ is the coordinate of each pixel center, $\mathbf{p_i}$ is the coordinate of each ground-truth vertex.

Using Eq. (3) for each location of ground-truth building vertices, we calculate the Gaussian density and generate the final ground-truth map. By taking the maximal response value, each peak is retained. Note that the center of the pixel, where the ground-truth vertex falls into, is not necessarily the location of the peak. The peaks always coincide with ground-truth building vertices locations. Theoretically, this setting allows for sub-pixel accuracy.

With these two learning targets, the building segments and vertices detection problems can now be solved in a regression manner.

3) Proposed double-branched network: Building segments and building vertices are predicted in parallel using two prediction heads. The TSD regressor outputs predicted TSD map, which contains intrinsically information of both building vertices and boundaries. These two kinds of information are extracted by two branches, i.e. Gaussian density regressor and outline refiner. With concatenation of the extracted features, the two branches have the chance to reuse the features that are helpful for predicting TSD map, which potentially improve the final predictions.

B. Vertex aided building polygonization

The predicted segmentation map is still not ideal for direct polygonization, due to building parts in shadows, confusion with background, and irregular shapes. Thus, in this paper, we propose a novel yet simple algorithm to generate building polygons automatically, based on the predicted segmentation map and Gaussian density map.

The predicted segmentation map is first binarized, so that each building instances are well separated. This threshold should be large enough to avoid merging neighbouring buildings, but small enough to retain basic shapes of the buildings. Based on this binarized predicted segmentation map, the initial polygons are generated, with area filtering to get rid of noise.

Our goal at this stage is to generate regularized building polygons based on the predicted Gaussian density map and the initial polygons. In order to make the final polygons as close as possible to manually delineated polygons, we adjust the initial polygons according to the predicted Gaussian density response. For each vertex in the initial polygon, we inspect a surrounding circular area with a certain radius, and replace each vertex with the location of highest Gaussian density response. If this response is smaller than a threshold, then we discard the corresponding initial polygon vertex. The output after removing redundant vertices is the final predicted building polygon.

TABLE I: Evaluation metrics

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR	AR_S	AR_M	AR_L
Mask R-CNN [18]	41.9	67.5	48.8	12.4	58.1	51.9	47.6	18.1	65.2	63.3
PANet [19]	50.7	73.9	62.6	19.8	68.5	65.8	54.4	21.8	73.5	75.0
PolyMapper [15]	**55.7**	**86.0**	**65.1**	**30.7**	68.5	58.4	**62.1**	39.4	75.6	75.4
VABP	48.2	75.7	54.5	29.8	**70.4**	**81.1**	61.6	**44.7**	**78.5**	**88.6**

III. Experiment

A. Dataset and experiment setup

We used the CrowdAI dataset as training and testing dataset [20]. This dataset contains satellite images of size 300×300 as RGB images, as well as annotations of building footprints. In this paper, we used the training dataset which contains in total 280,741 image tiles, and testing dataset with 60,317 image tiles.

In order to avoid padding in down-sample and up-sample path, we resized each image to spatial dimension 224×224. The annotations were re-calculated using linear transformation. The ground-truth TSD maps and Gaussian density maps were generated using Eq. (1) and Eq. (3), respectively.

B. Implementation and network training

The loss function for the TSD regressor $\mathcal{L}_T$ was chosen to be a pixel-wise mean squared error (MSE) loss, which is suitable for regression problems. To deal with high imbalance in Gaussian density regression, we used weighted MSE loss for the Gaussian density regressor, defined as

$$\mathcal{L}_G = \sum_{i,j} \left(\frac{\delta\left(t_{ij}^b - 1\right)}{N_1} \left(\hat{t}_{ij} - t_{ij}\right) + \frac{\delta\left(t_{ij}^b\right)}{N_0} \left(\hat{t}_{ij} - t_{ij}\right) \right), \tag{4}$$

where N_0 and N_1 are the total number of zeros and positive samples respectively, $\hat{t}$ and t are prediction and target respectively, t^b is the binarized Gaussian density map thresholded at 0.

The binary cross-entropy loss (BCE) was chosen to be the loss function for the outline refiner $\mathcal{L}_O$. The whole network is trained in an end-to-end manner with total loss

$$\mathcal{L} = \alpha\mathcal{L}_T + \beta\mathcal{L}_G + \gamma\mathcal{L}_O, \tag{5}$$

where α, β and γ are empirically selected weights for each loss.

C. Evaluation

The evaluation of the predicted building polygons were carried out by calculating the intersection over union (IoU), which is the ratio of the overlap area and the union area with regard to the ground-truth building polygons.

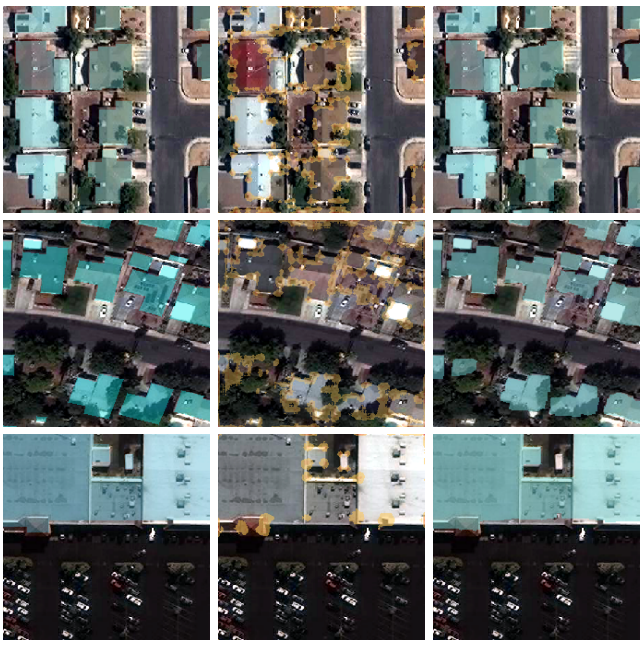
In order to better compare with other methods, we also calculated the average precision (AP) and average recall (AR) according to MS COCO [21].

IV. Results and discussion

A. Quantitative evaluation

Table I reports the AP and AR at IoU thresholds from 0.5 to 0.95 with step 0.05. We compare our method VABP to three published methods. We found out that our method works better for larger objects than smaller objects. We suspect that the TSD helps extract large footprints but tends to blend out small objects, as is shown in Eq. (1). When the building is too small, the surrounding area has more non-zero values than the building itself, which leads to potential false negative. By comparing to PolyMapper [15], our method is capable of processing a whole image scene, and has the advantage of fast inference, since we do not have multiple stages to predict each building vertex sequentially.

B. Qualitative evaluation



(a) (b) (c)

Fig. 3: Qualitative results. Column a: ground truth. Column b predicted Gaussian response. Column c: predicted building polygons.

In Fig. 3, three image scene examples are shown. The first row shows qualitative results of our method in easy scene, where buildings are all visible. The second row shows the performance in case of occlusion. The last row shows the performance for buildings with inner "holes". In general,

our method produces visually satisfying results with regular polygons, and works relatively well even in occlusion. Additionally, since we connect the vertices in a guided way, we do not have falsely generated polygons as reported in [7].

V. Conclusion

In this paper, we propose a vertex aided building polygonization method, which generates building polygons directly from satellite images. Our method has good performance with less model complexity, and is capable of processing image scenes with multiple buildings. We will work further on this method, e.g. testing with other satellite images in different countries, improving location accuracy of the extracted vertices.

References

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] K. Zhao, J. Kang, J. Jung, and G. Sohn, "Building extraction from satellite images using mask r-cnn with building boundary regularization," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 247–251.

[4] J. Mahmud, T. Price, A. Bapat, and J.-M. Frahm, "Boundary-aware 3d building reconstruction from a single overhead image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 441–451.

[5] W. Zhao, C. Persello, and A. Stein, "Extracting planar roof structures from very high resolution images using graph neural networks," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 187, pp. 34–45, 2022.

[6] F. Alidoost, H. Arefi, and M. Hahn, "Y-shaped convolutional neural network for 3d roof elements extraction to reconstruct building models from a single aerial image.," *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 5, no. 2, 2020.

[7] S. Zorzi, S. Bazrafkan, S. Habenschuss, and F. Fraundorfer, "Polyworld: Polygonal building extraction with graph neural networks in satellite images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1848–1857.

[8] S. Zorzi, K. Bittner, and F. Fraundorfer, "Machine-learned regularization and polygonization of building segmentation masks," in *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 3098–3105.

[9] W. Liu *et al.*, "Accurate building extraction from fused dsm and uav images using a chain fully convolutional neural network," *Remote Sensing*, vol. 11, no. 24, p. 2912, 2019.

[10] W. Sun and R. Wang, "Fully convolutional networks for semantic segmentation of very high resolution remotely sensed images combined with dsm," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 3, pp. 474–478, 2018.

[11] N. Girard, D. Smirnov, J. Solomon, and Y. Tarabalka, "Polygonal building extraction by frame field learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5891–5900.

[12] K. Wang and J.-M. Frahm, "Single view parametric building reconstruction from satellite imagery," in *2017 International Conference on 3D Vision (3DV)*, IEEE, 2017, pp. 603–611.

[13] Y. Qian, H. Zhang, and Y. Furukawa, "Roof-gan: Learning to generate roof geometry and relations for residential houses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2796–2805.

[14] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler, "Annotating object instances with a polygon-rnn," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5230–5238.

[15] X. Liu, G. Zhu, and X. Li, "Topological relationship extraction by two improved image segmentation methods," in *Geo-Informatics in Resource Management and Sustainable Ecosystem*, F. Bian, Y. Xie, X. Cui, and Y. Zeng, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 543–552, ISBN: 978-3-642-45025-9.

[16] Y. Wang, S. Zorzi, and K. Bittner, "Machine-learned 3d building vectorization from satellite imagery," *CoRR*, vol. abs/2104.06485, 2021. arXiv: 2104.06485. [Online]. Available: https://arxiv.org/abs/2104.06485.

[17] R. Bahmanyar, E. Vig, and P. Reinartz, "Mrcnet: Crowd counting and density map estimation in aerial and ground imagery," *CoRR*, vol. abs/1909.12743, 2019. arXiv: 1909.12743. [Online]. Available: http://arxiv.org/abs/1909.12743.

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[19] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768. DOI: 10.1109/CVPR.2018.00913.

[20] S. P. Mohanty *et al.*, "Deep learning for understanding satellite imagery: An experimental survey," *Frontiers in Artificial Intelligence*, vol. 3, 2020.

[21] T.-Y. Lin *et al.*, *Microsoft coco: Common objects in context*, 2014. DOI: 10.48550/ARXIV.1405.0312. [Online]. Available: https://arxiv.org/abs/1405.0312.