Rectilinear Building Footprint Regularization Using Deep Learning

Philipp Schuegraf[1] *, Zhixin Li[2], Jiaojiao Tian[1], Jie Shan[2], Ksenia Bittner[1]

[1]Remote Sensing Technology Institute, German Aerospace Center (DLR), Weßling, Germany
{philipp.schuegraf, jiaojiao.tian, ksenia.bittner}@dlr.de
[2]School of Civil Engineering, Purdue University, West Lafayette, Indiana,USA
{li2887, jshan}@purdue.edu

Commission II, WG II/4

KEY WORDS: Building instance segmentation, building vectorization, semantic segmentation, urbanization, aerial imagery.

ABSTRACT:

Nowadays, deep learning allows to automatically learn features from data. Buildings are one of the most important objects in urban environments. They are used in applications such as inputs to building reconstruction, disaster monitoring, city planing and environment modelling for autonomous driving. However, it is not enough to represent them in raster format, since applications require buildings as polygons. We use an existing, learning based approach to extract building footprints from ortho imagery and digital surface model (DSM) and propose a pipeline for building polygon extraction, which we call primary orientation learning (POL). The first step is to extract initial polygons, that contain a vertex for each pixel in the boundary of the footprint. Afterwards, the two primary orientation angles are regressed continuously. Using these orientation, we insert vertices such that all consecutive edges are perpendicular. To the best of our knowledge, our approach is the first to predict a continuous orientation angle for building boundary regularization. Furthermore, the proposed method is highly efficient with an average processing time of 2.879 ms for a single building.

1. INTRODUCTION

1.1 Problem Statement

In applications such as building reconstruction, disaster monitoring, city planning and environment modelling for autonomous driving, building footprints are crucial. Most works on building footprint extraction produce raster outputs, whereas applications require them in vector format. A robust approach to obtain buildings in vector format is to first predict raster buildings using a neural network and then applying postprocessing that outputs polygons. The results achieved by conventional methods are either limited in terms of generalization capacity (Zebedin et al., 2008; Cui et al., 2012; Tian and Reinartz, 2013) or are not restricted sufficiently to prior knowledge of regularity (Marcos et al., 2018; Gur et al., 2019; Hatamizadeh et al., 2020; Zhao et al., 2021; Zorzi and Fraundorfer, 2023).

Our goal is to achieve regularized building polygons, assuming that all vertices of buildings should have a rectangular angle. Whereas this assumption not always holds, it is sufficient for most buildings, especially residential houses and industrial buildings. A 90° angle always needs a reference axis, which is the primary orientation of a polygon. Even though for perfectly regularized polygons, in most cases the primary orientation is that of the longest sidelength, we are dealing with irregular polygons. Deep learning allows us to automatically learn features from a large training dataset. First, we use deep learning to extract building footprints from ortho imagery and photogrammetric digital surface model (DSM). Next, with our regularization framework called primary orientation learning (POL), we train a 1D convolutional neural network (CNN), from whose output we compute the primary orientation angle in continuous space. Subsequently, we use a learning free and iterative approach to insert vertices that make the initial polygon rectilinear, i.e. having 90° angles at every vertex. Figure 1 shows an example of a regularized building outline obtained by our method.

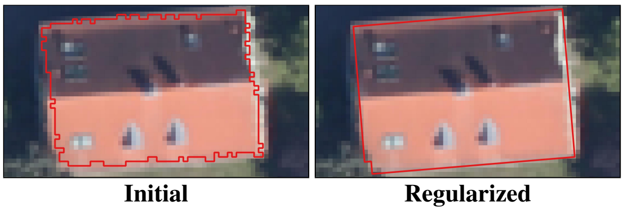

Figure 1. A building boundary regularization example in our test region, Braunschweig, Germany. The left figure represents the initial vectorization of the building footprint and the right one is the final regularized building boundary.

1.2 Related Work

Some research has been carried out on building footprint regularization. Most relevant to this paper are two works: The first one is that of Li et al. (2019), which inspired us to utilizing the primary orientation angle together with their simple yet effective rectilinearization algorithm. However, the way they compute the primary orientation angle is not robust. It relies on the minimal point density in the directions of the x and y axis after rotating the initial polygon by a candidate angle. But along the primary orientation direction, there may be an arbitrary number of vertices, depending on the roughness of the initial polygon. The second important work is frame field learning (Girard et al., 2021), where the idea of orientation is generalized to each pixel at the border of a building. The border orientation and its perpendicular direction are predicted along with the building footprint by a neural network. In a post-processing step, a

* Corresponding author

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-2-2024-217-2024 | © Author(s) 2024. CC BY 4.0 License.

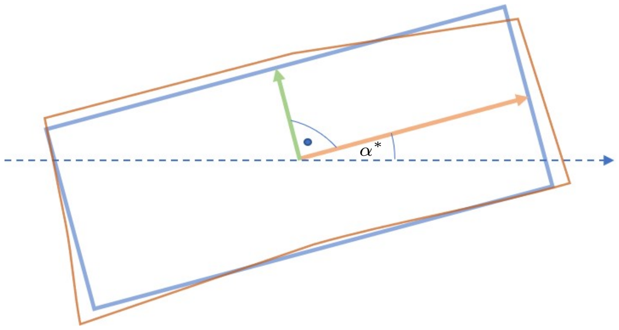

Figure 2. Visualization of a regular polygon (blue), an irregular polygon (brown), the primary orientation axis (orange) and the secondary orientation axis (green).

polygon is fit with vertices that align with the predicted border orientations. This allows to obtain regular polygons, even in complex cases. However, frame field learning does not guarantee rectilinearity, as Li et al. (2019) does.

Other work that includes the regularization of building outlines is that of Zebedin et al. (2008), where initial lines are filtered by forming a histogram of orientation and then removing outliers. The filtered line directions are used to reconstruct the building with regular appearance. This approach is flexible, as it is not restricted to 90° angles. Cui et al. (2012) use Hough transform to group an initial set of line segments into two perpendicular sets of parallel lines to represent the building boundary. Using those lines to construct an initial graph, edges in low-contrast regions are removed, since they do not represent building edges. Searching for cycles in the graph, the final building boundary is determined. This approach relies on the completeness of the initial line detection and is restricted to rectangular buildings. Tian and Reinartz (2013) also use Hough transform and intersection of line segments to form building boundaries, but allow two arbitrary main orientation directions.

More recently, end-to-end deep learning approaches have been utilized to improve building outline regularization. Marcos et al. (2018) propose to learn the parameterizations of active contour models to refine initial building blobs to regular polygons. Gur et al. (2019) came up with end-to-end trainable pipeline that iteratively updates an initial set of points similar as in active contour models. However, the predicted polygon is not necessarily reguarlized. Similar to Marcos et al. (2018); Gur et al. (2019), Hatamizadeh et al. (2020) proposes an active contour model based building boundary extraction which is end-to-end trainable and extents these capabilities to many arbitrary buildings in a patch, since initial contours are predicted by a CNN. Zhao et al. (2021) improve PolyMapper, which uses a recurrent neural network (RNN) to predict vertices recursively. Zorzi and Fraundorfer (2023) propose Re:PolyWorld, which is a multi-stage end-to-end trainable deep learning framework that predicts initial vertices, refines them and finally connects them to form polygons. It manages to score state-of-the-art (SOTA) metrics on the CrowdAI (Mohanty et al., 2020) building segmentation challenge.

2. METHOD

To obtain regularized building footprints, two steps are required. In the first step, we trained a neural network to extract building footprints. In the second step, we extracted building border pixels to form a polygon, followed by predicting the corresponding primary orientation, and then applied our rectilinearization algorithm to obtain regular polygons.

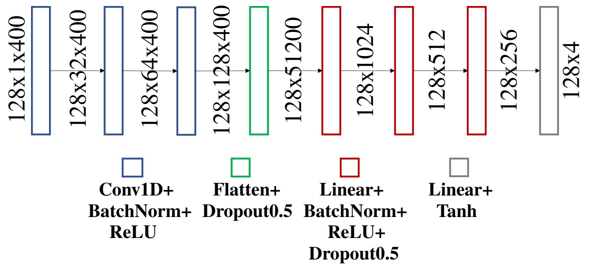

Figure 3. Visualization of the architecture of our proposed POL network. Convolutional layers extract local features for every vertex and linear layers compute global features and the regression output.

2.1 Building Footprint Extraction

We closely followed our previous work (Schuegraf et al., 2023) to obtain building instances. We trained UNet to predict a 3-class segmentation map that allows it to obtain building sections even if they are directly neighboring, using a post-processing step based on the watershed transformation. On the contrary, in this work we merge the obtained instances, since our aim is to regularize building blocks. We trained a UResNet34 with two ResNet34 encoders, where one encoder received an RGB patch and the other a DSM patch as input. At each level of resolution the feature maps of the encoder were merged by summing them. Then, the merged feature maps were passed to the decoder by providing them to the corresponding level of resolution. These so called skip-connection allow to gradually regain spatial resolution based on features, instead of unguided upsampling. The output of the UResNet34 were three logits from the same spatial resolution as the input patches. The three logits were input to the *softmax* function, which produces probability maps for three classes, which were background (0), building section (1) and separation line (2). Based on the probability maps, the *argmax* class was taken as the predicted class. Since we are interested in complete building blocks, we used both class 1 and 2 as the building class. To obtain instances, we used the watershed transform similar to Schuegraf et al. (2023), which, in this case without separation line, is equivalent to connected component analysis. Hence, we obtained a single instance for each building block.

2.2 Primary Orientation Learning: Vectorization and Regularization

The previously described method to obtain building footprints delivers them in raster format. To obtain instances, we first generate initial polygons and then refine them based on their primary orientation.

2.2.1 Initial Polygon Generation We applied a tree search to obtain an ordered set of boundary pixels, forming a polygon. This polygon has many redundant vertices and has irregular appearance, because of limited ground sampling distance (GSD) and imperfect building footprints. To remove many redundant vertices and simplify the polygon, we applied Douglas-Peucker (Douglas, 1973) with tolerance $\epsilon = 1.2$ m.

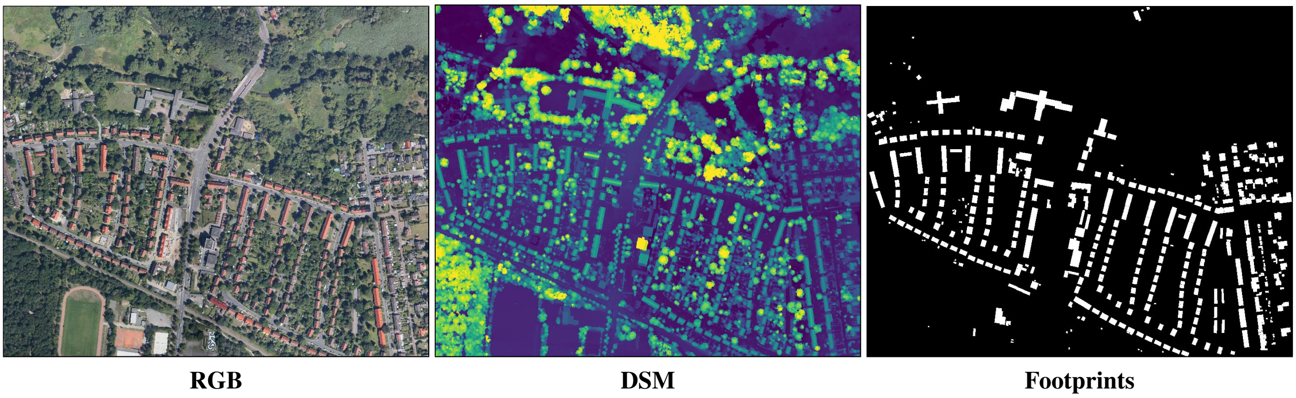

Figure 4. Our test area in Braunschweig, Germany. Three layers included within the data are shown, RGB image (left), DSM (middle), and the ground truth building footprints (right).

2.2.2 Direction Prediction & Rectilinearization The next step is based on the assumption, that the boundaries of a building are aligned with only two directions. We define the direction, along which the regular polygon stretches most, the primary orientation α^*. The 90° rotated primary axis is called secondary orientation $\beta^* = \alpha^* + 90°$. This can be seen in Figure 2, where the orange arrow represents the primary orientation axis, being rotated with respect to the blue, dashed arrow by angle α^*. For regular polygons, α^* can be obtained by computing the angle of the linesegment with the longest sidelength with respect to the positive x-axis (blue, dashed line). But for irregular polygons, the longest sidelength has no meaning. We represent a polygon $\mathcal{P} = [v_0, v_1, ..., v_{n-1}]$, which is a clockwise ordered set of n vertices $v_i \in \mathcal{V}$ with $\mathcal{V} = \{v_0, v_1, ..., v_{n-1}\}$ as a vector $\mathbf{p} = [x_0, y_0, x_1, y_1, ..., x_{n-1}, y_{n-1}, 0, ..., 0]^T$ with trailing zeros to bring each vector to the fixed length 400, which facilitates the length of all polygons in our dataset. We passed a minibatch of such vectors to a network consisting of 1D convolutional, rectified linear unit (ReLU), batch normalization, dropout and linear layers (see Figure 3). This network predicts the primary and secondary orientation angles $\hat{\alpha}$ and $\hat{\beta}$ by the parameters c_0 and c_2 of the complex polynomial

$$f(z) = z^4 + c_2 z^2 + c_0, \tag{1}$$

where

$$c_0 = u^2 \tag{2}$$

$$c_2 = -(u^2 + v^2) \tag{3}$$

$$\Leftrightarrow \begin{cases} u &= \sqrt{-\frac{1}{2}\left(c_2 + \sqrt{c_2^2 - 4c_0}\right)} \\ v &= \sqrt{-\frac{1}{2}\left(c_2 - \sqrt{c_2^2 - 4c_0}\right)}. \end{cases} \tag{4}$$

The ambiguity of the sign and order when regressing an angle directly is resolved in this representation of the orientation. We borrow this idea from Girard et al. (2021), where the coefficients are predicted at each pixel of an image along the boundary of buildings. However, we only predict a single complex value for each c_0 and c_2 for each polygon. Hence, we obtained 4 scalars for each input $\mathbf{p}$ from the network, from which we calculate the two complex numbers u and v using Equation (4). Then, we converted each of u and v into an angle with respect to the positive x-axis using trigonometry.

The network is trained using two loss functions, the first loss function

$$\mathcal{L}_{align} = |f(e^{i\theta^*}; \hat{c_0}, \hat{c_2})|^2, \tag{5}$$

where $\hat{c_0}$ and $\hat{c_2}$ are the predicted complex polynomial coefficients, enforces alignment of the prediction with the ground truth primary orientation angle θ^*. The second loss function

$$\mathcal{L}_{align90} = |f(e^{i\theta^{*T}}; \hat{c_0}, \hat{c_2})|^2, \tag{6}$$

enforces that the predicted secondary angle is aligned with $\theta^{*T} = \theta^* - \pi$. The total loss is

$$\mathcal{L} = \mathcal{L}_{align} + 0.2 \times \mathcal{L}_{align90} \tag{7}$$

Next, we applied the following rectilinearization algorithm for each of $\hat{\alpha}$ and $\hat{\beta}$, closely following Li et al. (2019):

1. Rotate the irregular polygon by $\hat{\theta} \in \{-\hat{\alpha}, -\hat{\beta}\}$
2. Given a clockwise ordered set of vertices $\mathcal{V} = \{v_0, v_1, ..., v_{n-1}\}$, where vertex v_i has coordinates (x_i, y_i), generate a line list $L = \{l_0, l_1, ..., l_{n-1}\}$;
3. Select the oblique line segments in L. Then for each oblique line segment $l_i \in L$,
    a. Calculate two candidate points to be inserted based on the two subsequent vertices v_i and v_{i+1}:

    $$v1_c = (x_i, y_{i+1})$$
    $$v2_c = (x_{i+1}, y_i)$$

    b. The relative position of each candidate point relative to l_i is determined using

    $$d1 = \begin{vmatrix} x_i & x_{i+1} & x_i \\ y_i & y_{i+1} & y_{i+1} \\ 1 & 1 & 1 \end{vmatrix},$$

    $$d2 = \begin{vmatrix} x_i & x_{i+1} & x_{i+1} \\ y_i & y_{i+1} & y_i \\ 1 & 1 & 1 \end{vmatrix},$$

    where $d1$ is the relative position of $v1_c$ and $d2$ that of $v2_c$.

    c. Since we are dealing with clockwise-oriented polygons, a negative $d1$ or $d2$ means that either $v1_c$ or

https://doi.org/10.5194/isprs-annals-X-2-2024-217-2024 | © Author(s) 2024. CC BY 4.0 License.

$v2_c$ is outside the polygon and hence is inserted into the polygon between v_i and v_{i+1}.

Since we applied the above algorithm twice for two different angles, we selected the rectilinear polygon that has the higher intersection over union (IoU) with the irregular polygon.

3. EXPERIMENTS

We carried out two experiments. Both experiments are based on the footprints from our raster footprint extraction method, trained according to Schuegraf et al. (2023). The first experiment is the baseline evaluating the method on our Braunschweig, Germany test region. The second experiment is our neural network based regularization on the same test region. See a visualization of the test area in Figure 4.

3.1 Baseline

The baseline method is that of Li et al. (2019). The main difference to our approach is that the baseline uses a learning free procedure to obtain the primary orientation angle.

3.2 Primary Orientation Learning

We trained the proposed POL network on a dataset consisting of 92600 regular building polygons from public sources of the cities of Berlin, Cologne and Hamburg, Germany, as well as Medellin, Columbia and validated after every epoch on 958 polygons of Cologne, Germany. Since the trained model should work on irregular polygons, we slightly shifted each vertex of the regular polygons by a 2D normal distribution centered at the original vertex position with standard deviation $0.5\,\mathrm{m}$. Additionally, we randomly rotated every polygon and adjusted the corresponding ground truth angle accordingly to increase the variety of training samples. We used the Adam optimizer with learning rate 0.001, batch size 128 and multiplied the learning rate by 0.9 after every ten epochs. We let the training run for 500 epochs and selected the model that performed best on the validation dataset.

We extracted the initial polygons from the predicted raster footprints by tracing the pixels along the boundary of each connected component. Then, we applied Douglas-Peucker with tolerance $1.2\,\mathrm{m}$ to simplify the initial polygon. We applied our trained POL network to the simplified polygon to obtain two orientation angles. Then, we applied the rectilinearization algorithm for each of the predicted angles and selected the polygon that has the larger IoU with the simplified polygon.

3.3 Evaluation

To judge the capability of our proposed method, we evaluated it on an RGB and DSM showing an area in Braunschweig, Germany. The data was captured by an aerial 3K camera at $0.1\,\mathrm{m}$ GSD and downsized to $0.3\,\mathrm{m}$ GSD.

Common metrics to evaluate building footprint quality are

$$IoU = \frac{TP}{TP + FP + FN}, \tag{8}$$

$$Prec = \frac{TP}{TP + FP}, \tag{9}$$

$$Rec = \frac{TP}{TP + FN}, \tag{10}$$

and

$$F1 = 2 \times \frac{Prec \times Rec}{Prec + Rec}, \tag{11}$$

where TP, FP, FN are the true positive, false positive and false negative of the building class. Additionally, we provide the inference time, training time and the angle prediction error

$$\varepsilon = |\hat{\theta} - \theta^*|. \tag{12}$$

For POL, we processed the polygons of the whole test area at once and divided the inference time by the number of polygons. The experiments were carried out on a server with an NVIDIA GeForce RTX 2080 Ti GPU with 11019 MB for the neural network inference and a Intel® Xeon® Gold 6230 CPU @ 2.10GHz for the baseline inference. The server has 504 GB working memory. To gain more insight into the results, we visualized both results next to the ground truth.

4. RESULTS

We listed the metrics of resulting building footprint quality and training/inference time in Table 1. It shows that both the baseline method and our proposed POL achieve very similar or almost identical results in IoU and F1, whereas POL has a higher precision and the baseline has a higher recall. The similarity in IoU and F1 are explained by the fact that we used the identical initial footprints and regularization does not have a large effect on these metrics. On the other hand, the baseline method tends to add the new vertices more on the outside of the ground truth polygon. These results show that our approach for footprint regularization is not worse than the baseline in terms of quality. This can be verified visually in Figure 6, where both the baseline and our method have perfectly regular appearance. In Figure 5, the high quality of most of the resulting building footprints is visualized. Although the satisfying overall result, we encountered some missing detections. Those are due to tiny building size, low contrast or lack of visibility in the RGB image, which makes it hard for the footprint predictor to recognize them. Furthermore, the baseline achieves an angular error ε of about 1.5° lower than our POL. POL predicts angles continuously which removes ambiguity from the angle prediction and avoids a method intrinsic error of up to 1.0°, which the baseline method includes. On the other hand, our learning based method was trained only on slightly alternations of the regular ground truth polygons, which leads to a domain gap between training and test polygons. Furthermore, we used the orientation of the longest side as the ground truth annotation, which is inaccurate in many cases but easy to obtain for large quantities of ground truth polygons. However, the error of 4.2447° is still very low, but the baseline needs to test 181 possible angles to achieve this results, which results in the inference time of 78.261 ms, whereas POL only needs 2.879 ms to infer a single primary orientation angle. This computational advantage can be explained by two reasons. The first is the aforementioned necessity of the

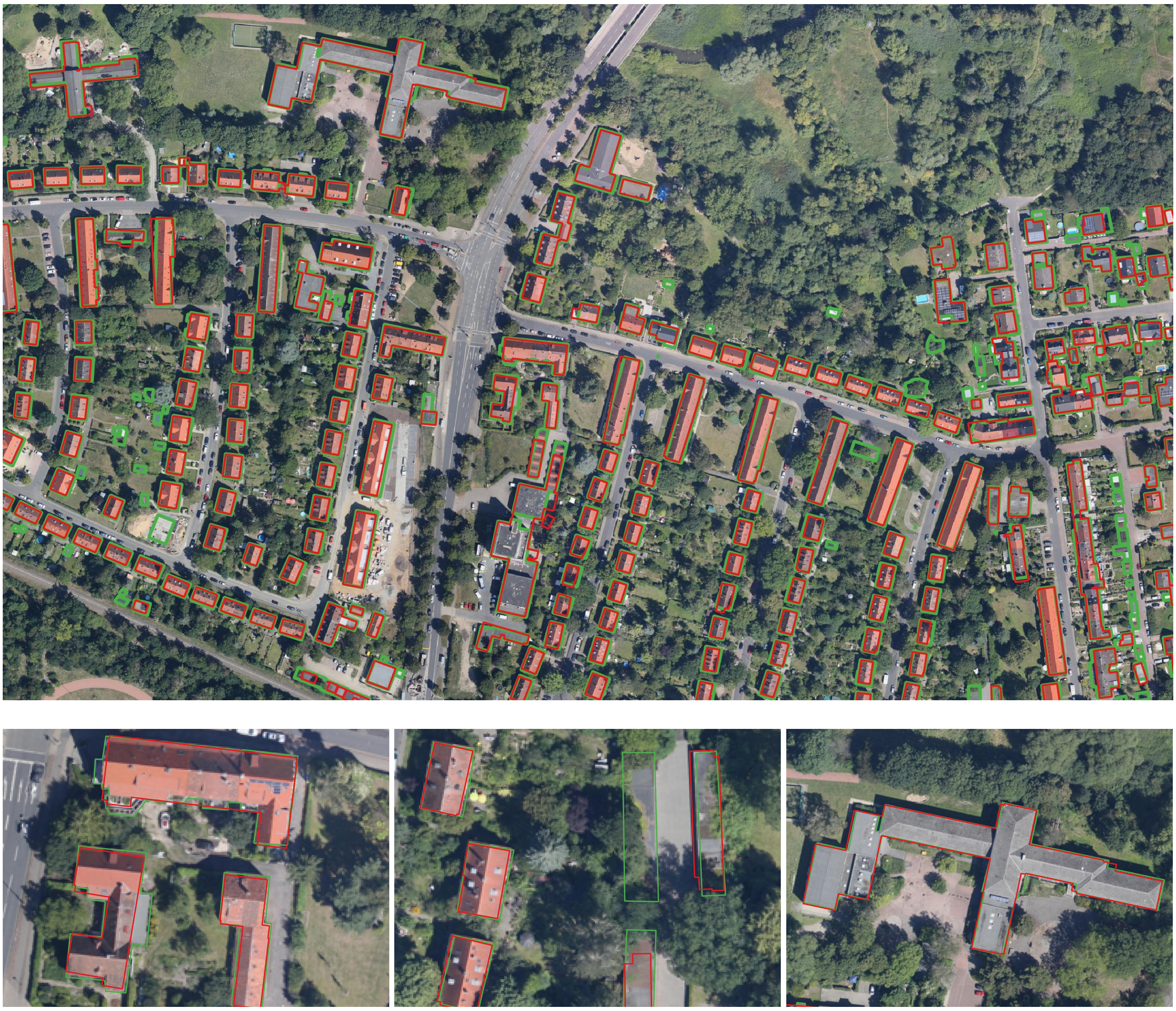

Figure 5. Our results in vector format on some part of the test area. Red polygons represent predicted building outlines, green polygons are ground truth polygons. The resulting polygons have regular shapes, i.e. right angles at every vertex with a low number of vertices. Even non-rectangular buildings are successfully regularized.

baseline to compute the axis density for 181 possible angles. The second one is the capability of batch processing in POL. POL can process the about 500 predicted initial polygons in the test area in a single forward pass in parallel.

5. CONCLUSION

We presented primary orientation learning (POL), a framework to predict real-valued, primary orientations of initial, irregular polygons in an end-to-end trainable manner. We leveraged those angles for accurate and efficient building polygon regularization, using a simple yet effective rectilinearization algorithm. Furthermore, we demonstrated the generalization capability of POL on polygons that are very different to those in the training dataset. Our analysis showed that our method achieves similar results as those of the reference method but overcomes the limitation of discrete valued angles. Since many buildings, especially in urban areas are not of rectangular structure, our future research will be centered on regularizing general buildings.

References

Cui, S., Yan, Q., Reinartz, P., 2012. Complex building description and extraction based on Hough transformation and cycle detection. *Remote Sensing Letters*, 3(2), 151–159.

Douglas, P., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2), 112 ff.

Girard, N., Smirnov, D., Solomon, J., Tarabalka, Y., 2021. Polygonal Building Extraction by Frame Field Learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5891 ff.

Gur, S., Shaharabany, T., Wolf, L., 2019. End to end trainable active contours via differentiable rendering. *arXiv preprint arXiv:1912.00367*.

Hatamizadeh, A., Sengupta, D., Terzopoulos, D., 2020. End-to-end trainable deep active contour models for automated image segmentation: Delineating buildings in aerial imagery.

Table 1. Quantitative evaluation of our method (POL) and the baseline method. We jointly evaluated quality and efficiency metrics. The baseline method for building regularization does not rely on machine learning, hence it has no training time.

Exp.	IoU	F1	Prec	Rec	Inf. Time	Train. Time	ε
Basline	0.7946	0.8855	0.8861	0.8850	78.261 ms	-	2.7355°
POL	0.7940	0.8852	0.9056	0.8657	2.879 ms	01:38:57 (hh:mm:ss)	4.2447°

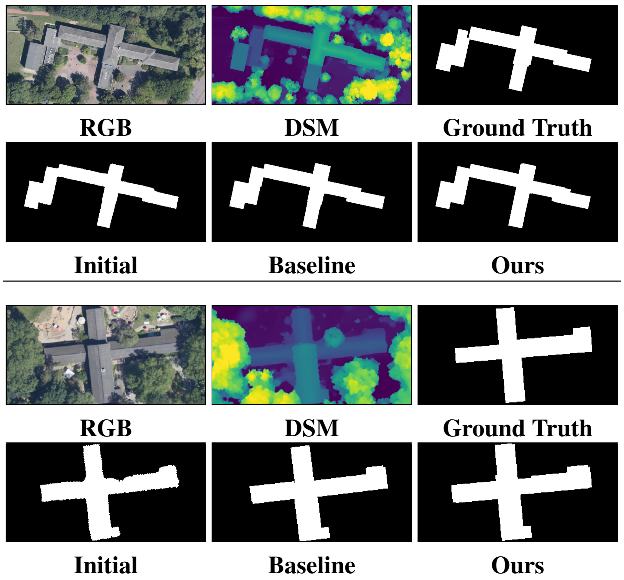

Figure 6. Visual result of two buildings in our test region. The baseline and our result are rectangular at every vertex, whereas the initial segmentation has irregular appearance. The horizontal line splits two different cases.

Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16, Springer, 730–746.

Li, Z., Xu, B., Shan, J., 2019. Geometric Object Based Building Reconstruction from Satellite Imagery Derived Point Clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W13, 73ff.

Marcos, D., Tuia, D., Kellenberger, B., Zhang, L., Bai, M., Liao, R., Urtasun, R., 2018. Learning deep structured active contours end-to-end. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8877–8885.

Mohanty, S. P., Czakon, J., Kaczmarek, K. A., Pyskir, A., Tarasiewicz, P., Kunwar, S., Rohrbach, J., Luo, D., Prasad, M., Fleer, S. et al., 2020. Deep Learning for Understanding Satellite Imagery: An Experimental Survey. *Frontiers in Artificial Intelligence*, 3.

Schuegraf, P., Zorzi, S., Fraundorfer, F., Bittner, K., 2023. Deep Learning for the Automatic Division of Building Constructions into Sections on Remote Sensing Images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 1-16.

Tian, J., Reinartz, P., 2013. Fusion of multi-spectral bands and dsm from worldview-2 stereo imagery for building extraction. *Joint Urban Remote Sensing Event 2013*, IEEE, 135–138.

Zebedin, L., Bauer, J., Karner, K., Bischof, H., 2008. Fusion of feature-and area-based information for urban buildings modeling from aerial imagery. *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part IV 10*, Springer, 873–886.

Zhao, W., Persello, C., Stein, A., 2021. Building outline delineation: From aerial images to polygons with an improved end-to-end learning framework. *ISPRS journal of photogrammetry and remote sensing*, 175, 119–131.

Zorzi, S., Fraundorfer, F., 2023. Re: Polyworld-a graph neural network for polygonal scene parsing. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16762–16771.