REGULARIZED BUILDING SEGMENTATION BY FRAME FIELD LEARNING

Nicolas Girard[1]*, Dmitriy Smirnov*[2]*, Justin Solomon*[2] *and Yuliya Tarabalka*[1,3]

[1]Université Côte d'Azur, Inria
[2]Massachusetts Institute of Technology
[3]LuxCarta Technology
email: nicolas.girard@inria.fr

ABSTRACT

We add a frame field output to an image segmentation neural network to improve segmentation quality and provide structural information for a subsequent polygonization step. A frame field encodes two directions up to sign at every point of an image. To improve segmentation, we train a network to align an output frame field to the tangents of ground truth contours. In addition to increasing performance by leveraging the multi-task learning effect, our method produces more regular segmentations with sharp building corners. GitHub: `github.com/Lydorn/Polygonization-by-Frame-Field-Learning`

***Index Terms*—** segmentation, regularization, remote sensing, frame field, PolyVector field

1. INTRODUCTION

Neural networks are now the go-to models for segmenting objects in remote sensing images because of their high learning capacity. For example, the fully-convolutional U-Net [1] is a popular model for pixel-wise classification, with favorable performance for building segmentation. Geographic information systems, however, require information to be in vector format (i.e., polygons and curves) instead of bitmap images, the output of most object segmentation networks.

A first approach is to vectorize the classification map of the network by, for example, using some contour detection (e.g. marching squares [2]) followed by a polygon simplification step (e.g., Ramer–Douglas–Peucker [3]). This approach yields mixed results when the classification map is not perfect, especially because the corners of buildings are rounded when using conventional deep learning methods for image segmentation. To produce cleaner buildings, one work uses two encoders: one shared decoder and one discriminator to learn how to regularize segmentation maps produced by another trained segmentation model [4]. The final loss is composed of the reconstruction cross-entropy loss, an adversarial loss and regularization losses that constrain the output map to align with the intensities of the input image. This method requires computing large matrices of pairwise discontinuity costs between pixels and involves adversarially training a system of networks which is less stable than conventional supervised learning.

Another approach is to directly learn an output in vector format. Curve-GCN [5] trains a graph convolutional network (GCN) to iteratively deform a polygon to fit the object centered on a patch. PolyMapper [6] uses a CNN-RNN to predict polygon vertices one at a time. However GCNs and RNNs are harder to train compared to CNNs.

In this paper, we introduce an algorithm for segmenting buildings that avoids the challenges above by adding a frame field output to a fully-convolutional network. Our work is closer to the first approach in that our network learns an intermediary grid-like representation of the segmentation which is then fed to an ad hoc polygonization method. As noted above, only learning the pixel-wise classification of objects makes the polygonization step challenging. We thus train the network to also learn a frame field aligned with the outline of objects. That frame field not only increases segmentation performance giving sharper corners but also provides useful information for the vectorization step. This paper focuses on the segmentation and frame field learning part and leaves the polygonization step as future work.

The main contributions of this paper are:

1. Learning a frame field aligned to object tangents, improving segmentation thanks to a multi-task learning effect.
2. Applying coupling losses between outputs so that they are coherent with one another, further leveraging multi-task learning.
3. Providing additional structural information in the form of a frame field for subsequent polygonization.

2. FRAME FIELDS

Our method is inspired by vectorization of line drawings by optimizing a frame field [7]. A frame field is a 4-*PolyVector field* [8], locally two symmetric line fields with independent scales. Similarly to [7], we learn a smooth frame field such

Thanks to ANR for funding the project EPITOME ANR-17-CE23-0009 and to Inria Sophia Antipolis - Méditerranée "Nef" computation cluster for providing resources and support.

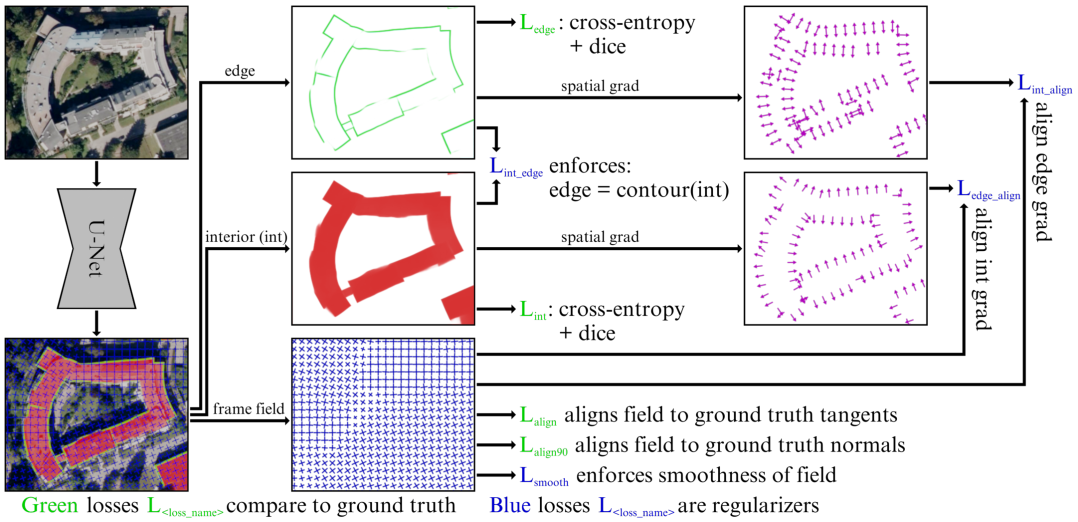

Fig. 1. Overview of the training procedure for our model that output segmentation maps as well as a frame field.

that at least one field direction is aligned to the polygon tangent at that location. Near corners of polygons, the field will be able align to *both* tangents on either side of the corner. Away from polygon boundaries, the frame field does not have any constraints to align to the ground-truth polygons. It is, however, trained to be smooth everywhere and not collapse into a simple line field. As with [7], we formulate computation of the field variationally. Unlike their approach, however, after discretization, we use a neural network to learn the field for every pixel of the image, an approach also explored in [9] for graphics applications.

The key to using frame fields efficiently is in their representation. For completeness, we present here relevant parts of [7, §3.1]. Suppose we are given two directions u, v representing our frame field at some point in the image. We identify the image plane with the complex numbers $\mathbb{C}$ and take $u, v \in \mathbb{C}$ as complex vectors. Consider the following complex polynomial $f(z)$:

$$f(z) = (z^2 - u^2)(z^2 - v^2) = z^4 + c_2 z^2 + c_0 \qquad (1)$$

The constants $c_0, c_2 \in \mathbb{C}$ determine u and v up to relabeling and sign: Every pair $(c_0, c_2) \in \mathbb{C}^2$ uniquely determines a frame $\{\pm u, \pm v\}$, agnostic to the labeling of u vs. v as well as their sign. We use $f(z; c_0, c_2)$ to denote the function above parameterized by the coefficients c_0 and c_2. Recovering the frame directions from (c_0, c_2) is equally straightforward:

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

The expression on the right is not unique as it corresponds to the symmetry of both directions.

Learning a (u, v) pair per pixel so that one of the vectors $\{u, -u, v, -v\}$ aligns to the ground truth tangent induces challenging issues involving labeling and sign. To avoid this complexity, we instead learn a (c_0, c_2) pair, which has no sign or ordering ambiguity.

3. METHOD

3.1. Starting with basic image segmentation

See Fig. 1 for an overview of the method. We use a U-Net [1] with a 2-channel output to segment both object interiors and contours, so that we can separate adjacent buildings, as in city centers. Annotation polygons are rasterized in an anti-aliased manner to pixel-wise ground truth weights: W_{int} for the interior of polygons and W_{edge} for the edges of polygons, which are given a width of a few pixels. We then use a linear combination of the cross-entropy loss and the Dice coefficient for the loss L_{int} applied on the interior output as well as the loss L_{edge} applied on the contour (edge) output.

3.2. Adding a frame field output

We then add another head to our network by adding a convolution that takes as input the concatenation of the segmentation output and the features of the layer before that. The output has four channels: two for c_0 and two for c_2. The ground truth is an angle $\theta_\tau \in [0, \pi)$ of the unsigned tangent vector of the polygon contour. We use three losses to train the frame field:

$$L_{align} = \frac{1}{HW} \sum_{\vec{x} \in I} W_{edge}(\vec{x}) |f(e^{i\theta_\tau}; c_0(\vec{x}), c_2(\vec{x}))|^2 \qquad (3)$$

$$L_{align90} = \frac{1}{HW} \sum_{\vec{x} \in I} W_{edge}(\vec{x}) |f(e^{i\theta_g}; c_0(\vec{x}), c_2(\vec{x}))|^2 \qquad (4)$$

$$L_{smooth} = \frac{1}{HW} \sum_{i=0,2} \sum_{\vec{x} \in I} ||\nabla c_i(\vec{x})||^2 \qquad (5)$$

Fig. 2. Crop of results on the sample test image available on the public leaderboard of the Inria dataset challenge.

Here, θ_w encodes the direction of a vector w, i.e., in complex language we can write $w = ||w||_2 e^{i\theta_w}$. H and W are the height and width of the image, resp.

L_{align} enforces alignment of the frame field with the tangent directions. This term is small when the polynomial $f(\cdot; c_0, c_2)$ has a root near $e^{i\theta_\tau}$, implicitly implying that one of the field directions $\{\pm u, \pm v\}$ is aligned with the tangent direction τ . Since (1) has no odd-degree terms, this term has no dependence on the sign of τ , as desired.

$L_{align90}$ prevents the frame field from collapsing into a line field by expressing a slight preference for the field to be also aligned with $g = \tau^\perp$.

L_{smooth} is a Dirichlet energy measuring the smoothness of the functions $c_0(\vec{x})$ and $c_2(\vec{x})$ as a function of the location $\vec{x}$ in the image. Smoothly-varying (c_0, c_2) pairs imply a smooth set of frame directions.

3.3. Adding coupling losses between different outputs

Given that our outputs should be highly correlated, we add coupling losses that force them to be mutually coherent:

- $\boldsymbol{L_{int\_align}}$: Aligns the gradient of the output interior map with the frame field (using an $align$ loss).
- $\boldsymbol{L_{edge\_align}}$: Aligns the gradient of the output edge map with the frame field (using an $align$ loss).
- $\boldsymbol{L_{int\_edge}}$: Makes the output edge map be equal to the output interior map gradient norm. This loss is not applied inside detected objects so that buildings can still be separated by the edge map.

4. EXPERIMENTAL SETUP

4.1. Training

We use the Inria Aerial Image Labeling Dataset [10] (*Inria dataset* from now on) for our experiments. It has 360 images of 5000×5000 px from 5 cities in Europe and USA. The train and test sets have 180 images each from different cities. We polygonized the ground truth masks with [11] in order to be able to compute the ground ground truth for the frame field, by rasterizing separately each edge in every polygon and giving the edge's angle as value for the drawn segment. The resulting gray image gives us for each pixel the θ_τ used in L_{align}.

We use a U-Net model with a ResNet-101 encoder, optimized using distance weights for the cross-entropy loss as is done for the original U-Net [1].

4.2. Results

We compare with the best method on the leaderboard[1]: ICTNet [12]. In order to show the detected contours, we apply marching squares [2] followed by Ramer-Douglas-Peucker [3] simplification with a tolerance of 1px. See Fig. 2 for results on the sample test image from the leaderboard. ICTNet has an overall mIoU of 80.3% while our Unet-ResNet101 with frame field learning has 74.1% (Unet-ResNet101 without frame field learning has 71.9%). We however obtain more regular contours. To quantify this regularity we measure the relative angle distribution of contour

[1] https://project.inria.fr/aerialimagelabeling/leaderboard/

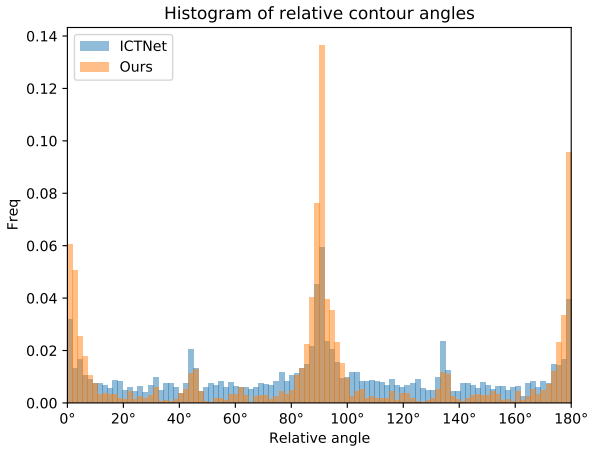

Fig. 3. Regularity measure on the sample test image. A more uniform histogram means less regular contours.

tangents for the same test sample, see Fig. 3. The relative angles are relative to the orientation of the minimum rotated rectangle of each building. On another note, annotations for the Inria dataset are not perfect, with building corners rarely well-localized because of misalignment. We hypothesize that is the reason even the best method on the dataset (ICTNet) outputs rounded corners: because that is the only way to maximize mIoU with the expected ground truth. Our network, being regularized with additional frame field losses, is able to output sharp corners. However that also means it cannot hope to achieve state-of-the-art performance if only mIoU on an imperfect ground truth is considered.

5. DISCUSSION AND CONCLUSION

We improve on the task of image segmentation by adding an output to the model, a frame field. Only a few extra convolutions are added, thus inference time is not affected. Because the network learns an additional highly correlated task, the segmentation performance is increased (see Table **??**). Additionally, the use of coupling losses between outputs forces them to be coherent with one another, makes the network more accurate and regularizes the segmentation (see Fig. 2 and Fig. 3).

This work is additionally an intermediate step toward improved polygonization of the segmentation maps, which can use the frame field output for additional structural information. For example, the frame field makes it easier to distinguish between corners of buildings and curves linking those corners forming a piecewise smooth shape.

Our approach is very efficient, since the trained model is a single fully-convolutional network that is optimized by local supervision signals: All outputs for a pixel only need image information in a neighborhood around that pixel, just like regular image segmentation. The training is straightforward, unlike adversarial training or direct shape regression by a network approaches, which require significant tuning to make them work. The frame field output can be dropped in any existing image segmentation network, as well as be used in a multi-class segmentation setting.

As future work, we will develop a polygonization method that will leverage both the segmentation and the frame field.

6. REFERENCES

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," *MICCAI*, 2015.

[2] William Lorensen and Harvey E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *SIGGRAPH*, 1987.

[3] Wikipedia contributors, "Ramer–douglas–peucker algorithm — Wikipedia, the free encyclopedia," 2019.

[4] Stefano Zorzi and Friedrich Fraundorfer, "Regularization of building boundaries in satellite images using adversarial and regularized losses," *IGARSS*, 2019.

[5] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler, "Fast interactive object annotation with curve-gcn," in *CVPR*, 2019.

[6] Zuoyue Li, Jan Dirk Wegner, and Aurélien Lucchi, "Topological map extraction from overhead images," *ICCV*, 2019.

[7] Mikhail Bessmeltsev and Justin Solomon, "Vectorization of line drawings via polyvector fields," *ACM Trans. Graph.*, 2019.

[8] Olga Diamanti, Amir Vaxman, Daniele Panozzo, and Olga Sorkine-Hornung, "Designing n-polyvector fields with complex polynomials," *Eurographics SGP*, 2014.

[9] Maria Taktasheva, Albert Matveev, Alexey Artemov, and Evgeny Burnaev, "Learning to approximate directional fields defined over 2d planes," *CoRR*, 2019.

[10] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez, "Can semantic labeling methods generalize to any city? the Inria aerial image labeling benchmark," in *IGARSS*, 2017.

[11] Nicolas Girard, Dmitriy Smirnov, Justin Solomon, and Yuliya Tarabalka, "Polygonal building segmentation by frame field learning," *ArXiv*, vol. abs/2004.14875, 2020.

[12] Bodhiswatta Chatterjee and Charalambos Poullis, "On building classification from remote sensor imagery using deep neural networks and the relation between classification and reconstruction accuracy using border localization as proxy," in *2019 16th Conference on Computer and Robot Vision (CRV)*. IEEE, 2019, pp. 41–48.