LineFit: A Geometric Approach for Fitting Line Segments in Images
-
Supplementary Material

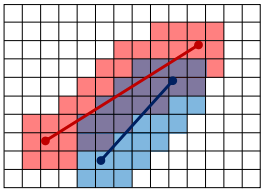
Marion Boyer[1,2], David Youssefi[2], and Florent Lafarge[1]

[1] Inria Centre Université Côte d'Azur, France
[2] Centre National d'Etudes Spatiales (CNES), France
Firstname.lastname@{inria,cnes}.fr

This supplementary material provides a description of (i) the evaluation metrics (Section 1) and of (ii) the regularization operator for orthographic projection images (Section 2), (iii) a presentation of the *RoofSat* dataset (Section 3), and (iv) additional results and comparisons (Section 4).

1 Details on evaluation metrics

Accuracy metrics. Our grid-based accuracy metrics include the popular Average Precision, Average Recall and Average F-score computed in a strict way, i.e. without diffusion or thresholding. This explains why scores given in Table 1 are relatively low on the two datasets. As illustrated in the inset, we simply operate an 8-connexity dilation on the pixels crossed by the detected line segments (blue line) and the Ground Truth line segments (red line) for discretization. The resulting occupancy areas (True Positive in purple, False Positive in blue and False Negative in red) allows us to compute Precision, Recall and F-Score by the traditional formulas.



Regularization metrics. The Degree of Freedom score (DoF) is defined as the ratio of the degree of freedom $k_{\mathbf{x}}$ of a configuration of line segments $\mathbf{x}$ to twice the number of line segments $|\mathbf{x}|$ in the configuration, multiply by a factor 100 to be expressed in percent:

$$\mathrm{DoF}(\mathbf{x}) = 100 \times \frac{k_{\mathbf{x}}}{2|\mathbf{x}|} \tag{1}$$

Having a DoF score of 100 means that all line segments of the configuration are independent and not subject to regularity constraints. Note that a DoF score of 0 is not possible as, even in the case where all the line segments were mutually co -linear, at least one of the line-segment must be described by two parameters. In such a case, the minimal score would be $1/|\mathbf{x}|$.

The degree of freedom $k_{\mathbf{x}}$ is computed via the counting of the number of line-segments in the different regularized groups. For orthographic projection images

(i.e. with parallelism, orthogonality and co-linearity as regularities), the degree of freedom is given by

$$k_{\mathbf{x}} = 2|\mathbf{x}| - \sum_{i=1}^{K^{\parallel,\perp}} (k_i^{\parallel,\perp} + 1) - \sum_{i=1}^{K^{\parallel}} (k_i^{\parallel} + 1) - \sum_{i=1}^{K^{--}} (k_i^{--} + 1) \tag{2}$$

where $K^{\parallel,\perp}$, $K^{\parallel}$ and K^{--} correspond to the number of parallel groups which are also mutually orthogonal, the number of parallel groups which are not orthogonal to other groups, and the number of co-linear sub-groups inside all the parallel groups respectively. $k_i^{\parallel,\perp}$ is the number of line segments in the i^{th} group of $K^{\parallel,\perp}$, $k_i^{\parallel}$ the number of line segments in the i^{th} group of $K^{\parallel}$, and k_i^{--} the number of line segments in the i^{th} group of K^{--}.

For perspective projection images (i.e. with concurrence to vanishing points and co-linearity as regularities), the degree of freedom is computed as

$$k_{\mathbf{x}} = 2|\mathbf{x}| - \sum_{i=1}^{K^{\star}} (k_i^{\star} + 1) - \sum_{i=1}^{K^{--}} (k_i^{--} + 1) \tag{3}$$

where $K^{\star}$ and K^{--} corresponds to the number of concurrent groups (which cannot be higher than the number of vanishing points) and to the number of co-linear sub-groups inside all the concurrent groups. $k_i^{\star}$ (respectively k_i^{--}) is the number of line segments in the i^{th} group of $K^{\star}$ (of K^{--} respectively).

In practice, the computation of $k_{\mathbf{x}}$ is performed under two tolerance thresholds for orthographic projection images, one in orientation set to 10^{-2} degree and one in distance also set to 10^{-2} pixel to absorb precision issues of the exact geometric intersection tests. For perspective projection images, we also introduce a tolerance threshold representing the maximal orthogonal distance between the supporting line and the vanishing point. We consider the supporting line passes through the vanishing point if the orthogonal distance is less than 0.05 pixels. This threshold is a bit higher than the distance threshold for orthographic projection to also absorb the imprecision of the positioning of the vanishing points. The use of these tolerance thresholds explains why the DoF scores of methods without regularization in Table 1 is not necessarily 100.

2 Regularity operator for orthographic projection images

For orthographic projection images, the global regularity operator enforces parallelism, orthogonality and co-linearity between the line segments of the configuration. We first group near parallel line segments by Mean-Shift on the supporting line orientations using a kernel width fixed to 5 degrees. We then connect these clusters into sub-graphs if they are near-orthogonal, i.e. if the orientation of the cluster centroids differs by 90 ± 5 degrees. Each sub-graph is constrained by a unique orientation. The latter is found by projecting inlier points of each cluster by the translation and rotation that send its center of mass to the origin and

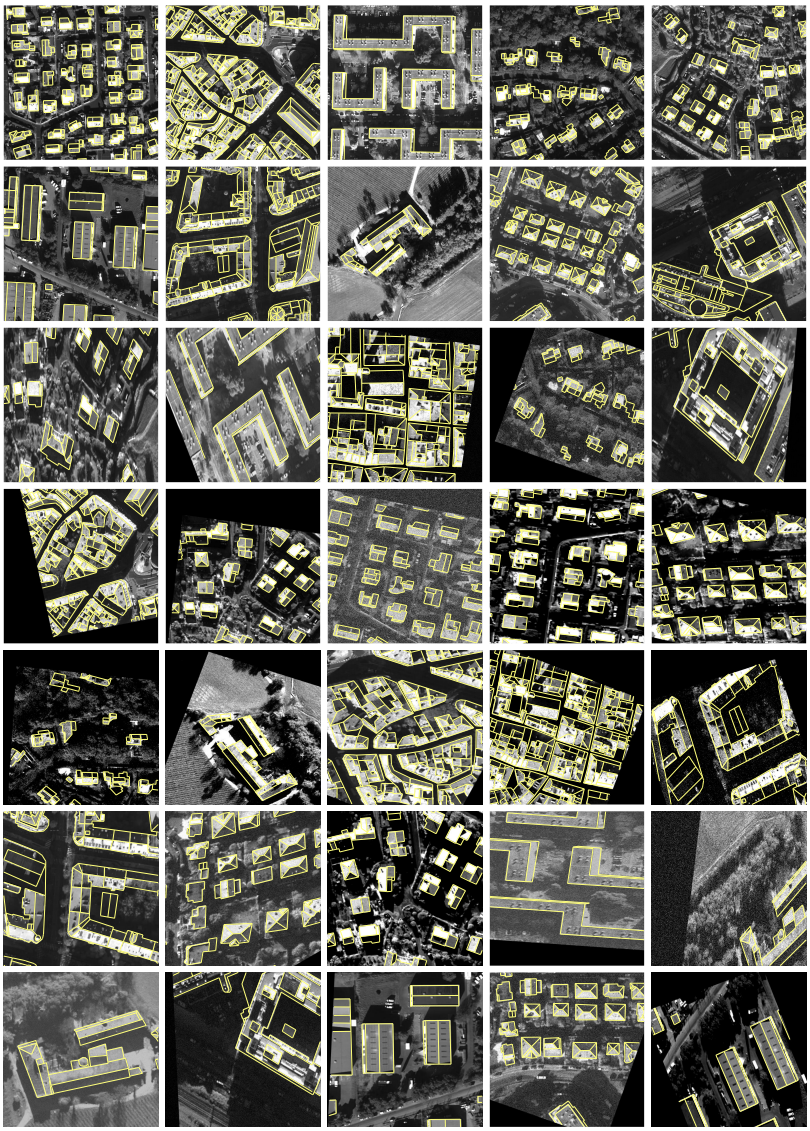


Fig. 1: *RoofSat* dataset. A sample of 35 images and their associated Ground Truth line segments are shown on different urban landscapes. The two top rows correspond to original satellite images; the five bottom ones incorporate geometric and radiometric transformations.

fitting the best weighted least square line on these projected inliers. The optimal supporting line is then projected back on each cluster centroid by the reverse transformation. Finally, near-colinear line segments are grouped by performing a Mean-Shift on the distance to the origin of the supporting lines in each parallel

cluster with a kernel width fixed to $\epsilon/2$, before re-positioning the supporting lines to their centroid.

3 Details on the *RoofSat* dataset

The *RoofSat* dataset is composed of 550 images of 550×550 pixels representing urban scenes at nadir. They originate from 11 PleiadesNeo satellite images at 30cm resolution, from which we perform data augmentation with rotation, scaling, translation, noise, blurring and contrast variation. Figure 1 shows a sample of 35 images and their associated Ground Truth line segments on various types of urban scenes, i.e. residential, dense downtown or industrial. Line-like structures in these images correspond to both contours and roof skeletons of buildings. Ground Truth capturing these line-like structures have been annotated by a remote sensing expert and are provided as a list of floating-precision line segments. This dataset is availabla at https://project.inria.fr/roofsat/.

4 Additional results and comparisons

We present here additional results on line segment detection and on application to object polygonalization.

Figures 2, 3 and 4 show some visual comparisons on images representing either purely organic shapes or man-made objects mixing curved shapes and line-like structures. In these two scenarios, our point-to-line fitting formulation is particularly competitive as it allows us to both accurately locate line segments on line-like structures and approximate curved shapes by chains of line segments. Note that LSD [3] scores are not included in the quantitative comparisons of Table 1 of the paper for space reasons. Its scores are similar to those of ELSED [9]. More specifically, AP, AR and AF scores of LSD are respectively 49.9, 36.7 and 41.8 on *YorkUrban* and 44.7, 32.4 and 33.3 on *RoofSat*.

Figure 5 shows the impact of various image gradient maps on the output line segment configuration. The gradient maps tested in this experiment include the basic one computed by finite differences, the surrogate gradient maps from the architectures DeepLSD [7] and Dexined [8], a gradient map generated using the popular segmentation method SAM [4], and, as a reference, the gradient map computed from the contours annotated by hand which are provided in the Berkeley segmentation dataset [6]. The basic gradient map and the DeepLSD-based one are computed in the same way than in the paper. For Dexined [8] and SAM [4], we threshold their contour probability maps at 0.5 to select our set of input points. For the Ground Truth annotation, we simply associate an input point to each contour pixel. DeepLSD-based gradient map is a good choice for a general description of the object whereas SAM-based gradient is particularly efficient on the silhouette of the object. However the quality of these gradient maps remain perfectible in comparison with the ground truth gradient map. Our result from the latter suggests than our fitting algorithm will benefit from the rapid progress in the field for generating more and more accurate gradient maps.

Figure 6 shows object polygonalization results at different levels of complexity. These levels are produced by connecting line segments detected at various approximation levels, ie with various parameter pairs (ϵ,σ). Figure 7 shows visual comparison of our method with PolyFormer [1] and Poly-RNN++ [5] on an organic (curved) object at different levels of complexity. Note that in all our experiments with PolyFormer and Poly-RNN++, we carefully set their additional inputs so that the returned polygons capture the objects of interest with almost no situation of under- or over-detection. More precisely, we performed for each image (i) a trial-and-error selection of the bounding-box or the query string to capture the objects of interest, (ii) a polygon accumulation with multiple runs in case of several disjoint objects, and (iii) no manual post-modifications of the polygons.

Figures 8, 9, 10 and 11 show visual results of object polygonalization on the *BSDS500* dataset when using our line segment detector with $\epsilon = 3$ and $\sigma = 5$. This setting is a good compromise between accuracy and complexity to capture both curved objects and line-like structures.

References

1. Acuna, D., Ling, H., Kar, A., Fidler, S.: Efficient interactive annotation of segmentation datasets with polygon-rnn++. In: CVPR (2018) 5, 10
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. PAMI **33**(5) (2011) 6, 9
3. Gioi, R.V., Jakubowicz, J., Morel, J.M., Randall, G.: Lsd: A fast line segment detector with a false detection control. PAMI **32**(4) (2010) 4, 6, 7, 8
4. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. In: ICCV (2023) 4, 9
5. Liu, J., Ding, H., Cai, Z., Zhang, Y., Satzoda, R.K., Mahadevan, V., Manmatha, R.: Polyformer: Referring image segmentation as sequential polygon generation. In: CVPR (2023) 5, 10
6. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV (2001) 4, 9
7. Pautrat, R., Barath, D., Larsson, V., Oswald, M.R., Pollefeys, M.: Deeplsd: Line segment detection and refinement with deep image gradients. In: CVPR (2023) 4, 6, 7, 8, 9
8. Soria, X., Sappa, A., Humanante, P., Akbarinia, A.: Dense extreme inception network for edge detection. Pattern Recognition **139** (2023) 4, 9
9. Suárez, I., Buenaposada, J.M., Baumela, L.: Elsed: Enhanced line segment drawing. PR **127** (2022) 4, 6, 7, 8
10. Xu, Y., Xu, W., Cheung, D., Tu, Z.: Line segment detection using transformers without edges. In: CVPR (2021) 6, 7, 8
11. Xue, N., Wu, T., Bai, S., Wang, F.D., Xia, G.S., Zhang, L., Torr, P.H.: Holistically-attracted wireframe parsing. In: CVPR (2020) 6, 7, 8

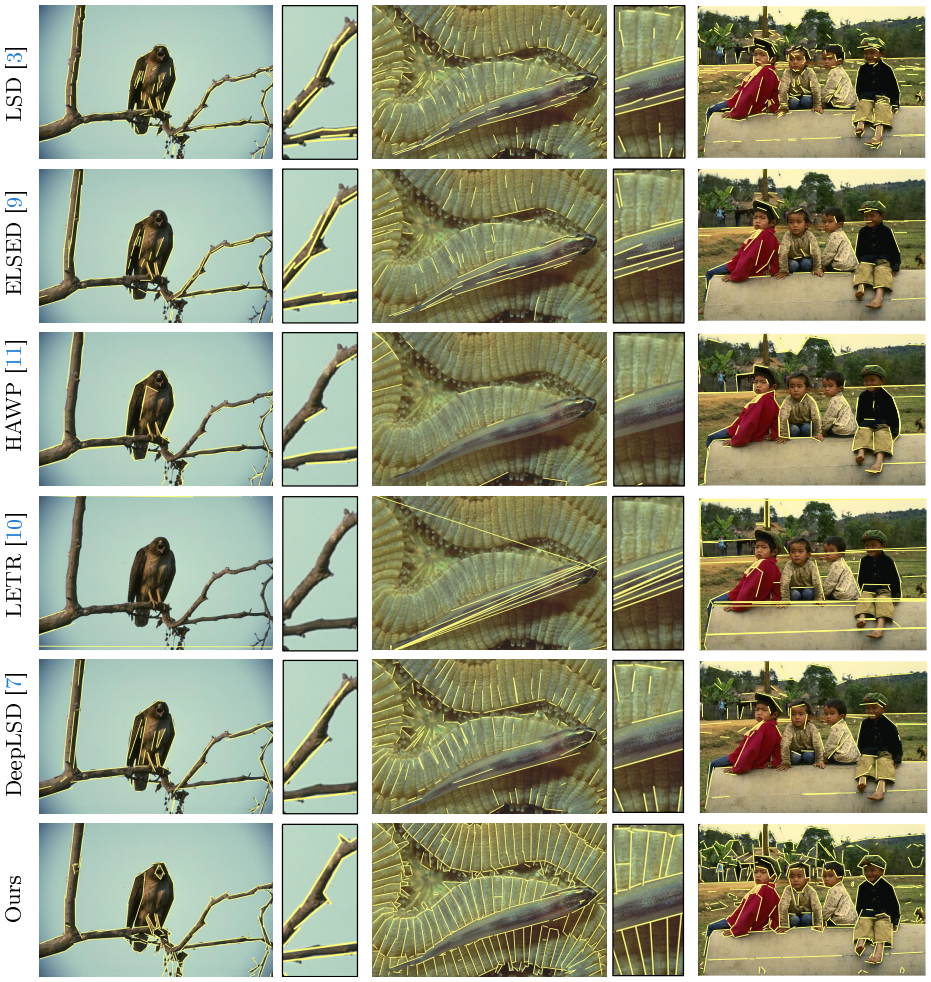


Fig. 2: Visual comparisons on organic objects. Our approach exhibits both a higher precision and a better completeness for approximating organic shapes by line segments. Learning-based methods LETR, HAWP, and, to a lesser extent, DeepLSD do not generalize well on such images. The traditional mechanisms LSD and ELSED generalize better but produce sparse configurations with frequent line segment overlapping (see close-ups). By setting the fitting tolerance parameter to a few pixels, our algorithm can absorb the noise around the contours of these organic shapes. This allows us to achieve simpler and more complete description of these objects. The images originate from the Berkeley segmentation dataset BSDS500 [2].

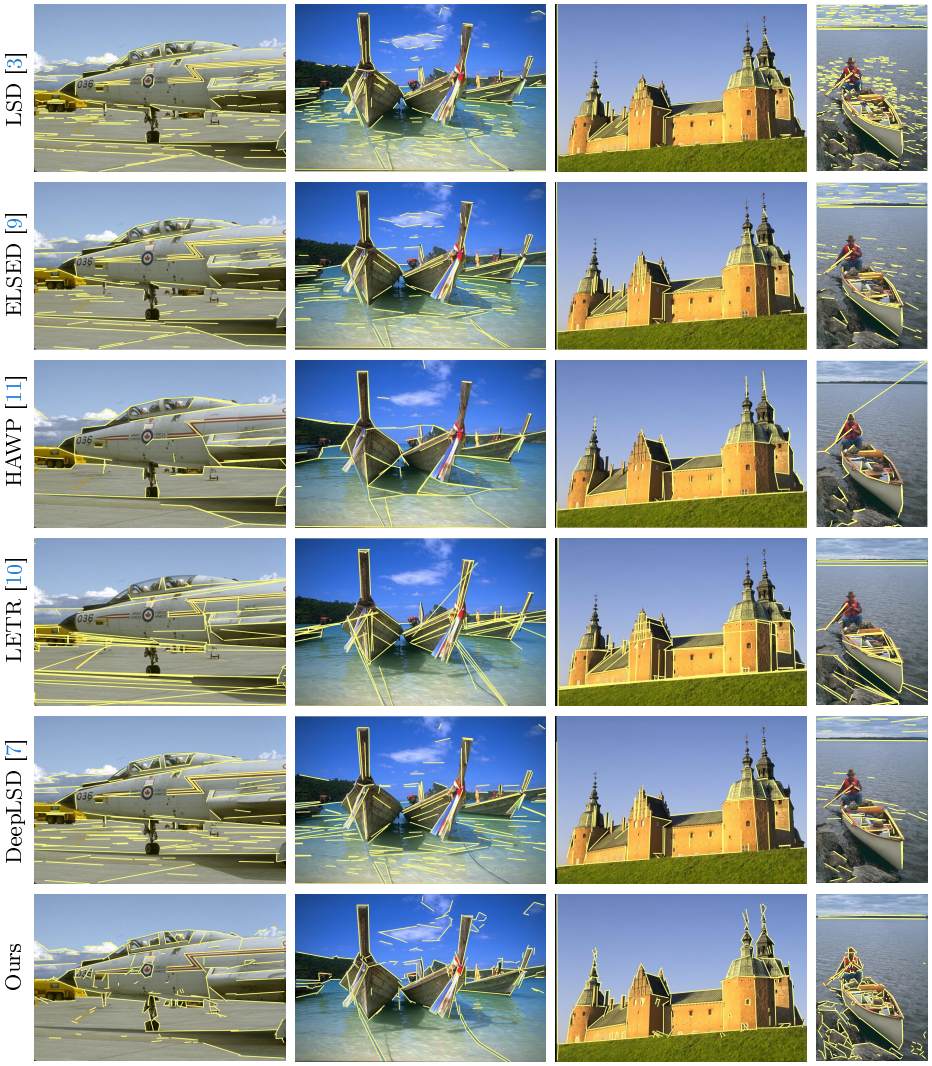


Fig. 3: Visual comparisons on man-made objects with curved shapes and line-like structures (part 1/2). Our algorithm performs best when images contain objects with both curved and line-like shapes. Our point-to-line fitting formulation allows us to accurately locate line segments on line-like structures while approximating curved shapes by chains of line segments. In particular, only our algorithm produces a fine and complete description of the jet fighter fuselage (left) or the mixed curved and piecewise-planar roof of the castle (3rd column).

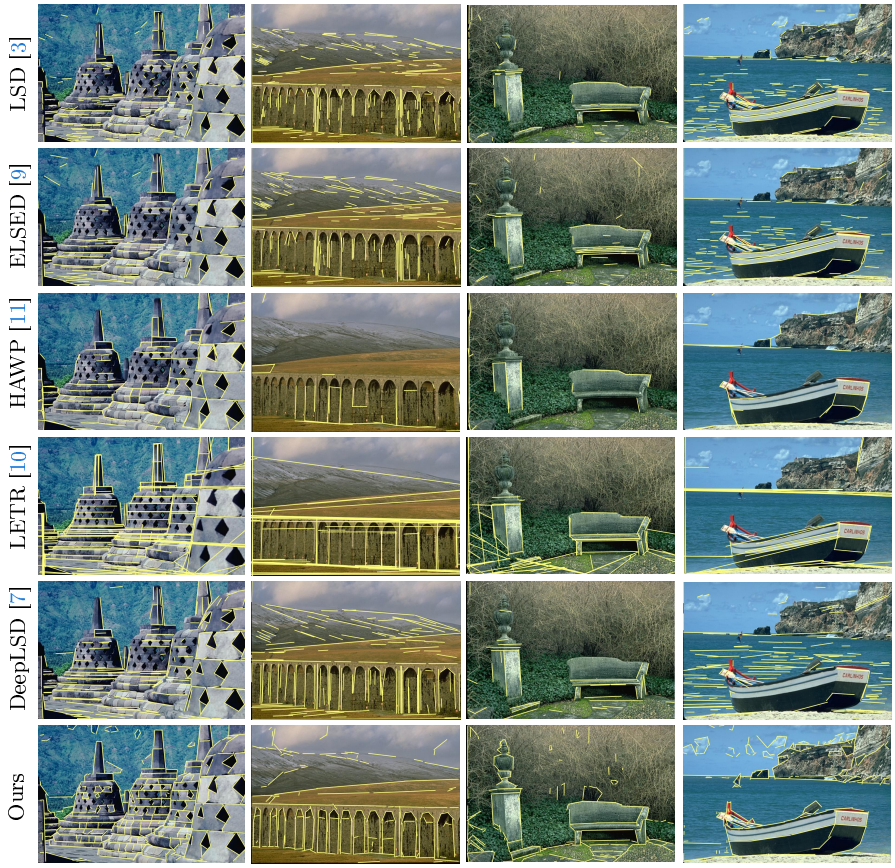


Fig. 4: Visual comparisons on man-made objects with curved shapes and line-like structures (part 2/2).

Fig. 5: Visual results from different image gradient maps. Using a basic gradient map computed by finite differences typically gives a sparse configuration of globally-accurate line segments (first column). Computing the image gradient map with DeepLSD [7] or Dexined [8] improves the quality of the configuration with more meaningful segments (second and third columns). Using the popular segmentation method SAM [4] to generate a gradient map gives good results to describe the silhouette of object but does not allow us to capture well the inside ornament details (fourth column). These different results can be compared to the configuration obtained when using a Ground Truth gradient map, i.e. a gradient map computed from the contours annotated by hand of the Berkeley segmentation dataset BSDS500 [2]. The latter configuration is complete and accurate both on silhouette and inside details (last column).

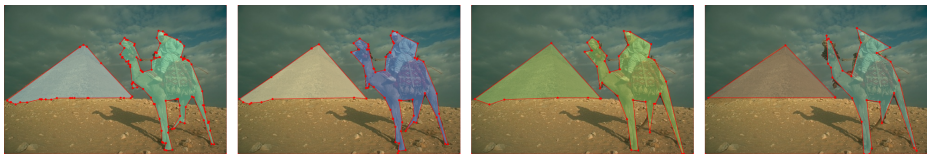


Fig. 6: Object polygonalization at different level of complexity. Detecting line segments at various approximation levels allows us to produce polygons with different number of edges. At high complexity (left), the curved objects, ie the camel and the person, are finely approximated while the line-like structure, ie the pyramid, is well captured despite a few extra vertices. At low complexity (right), the polygons roughly capture the curved objects but describe the line-like structure in a very compact manner.

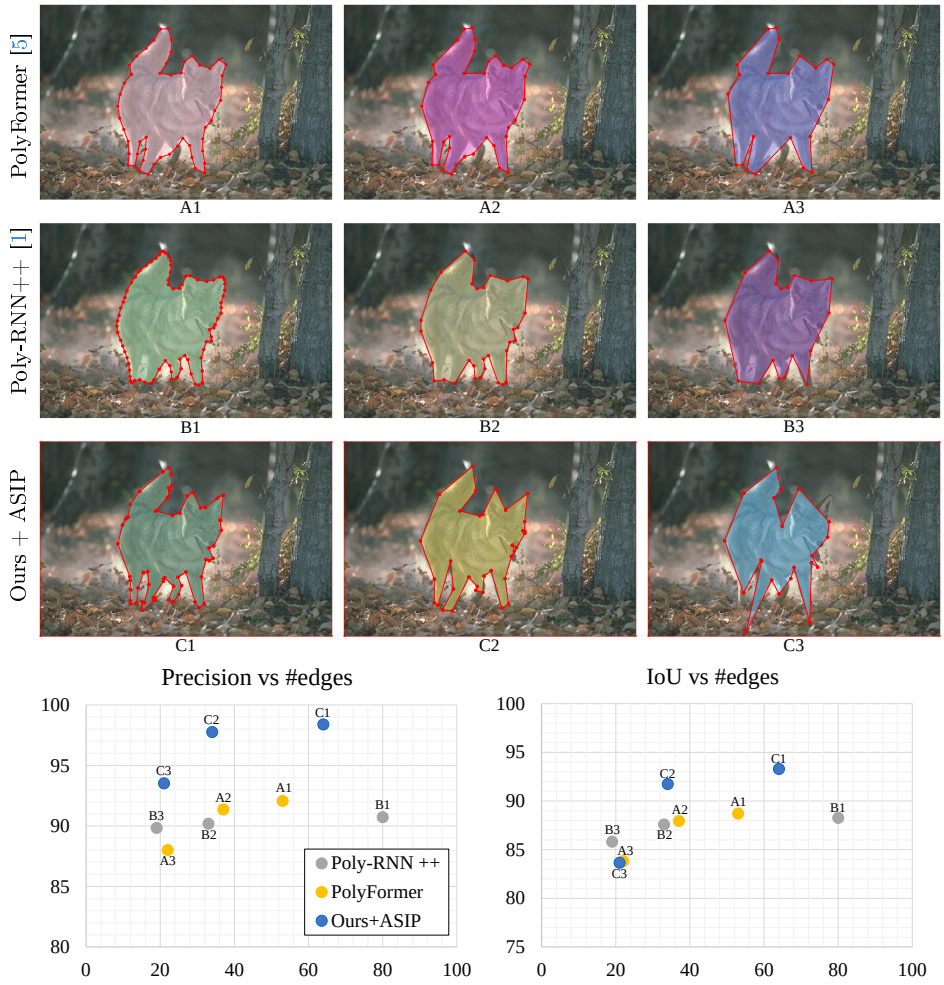


Fig. 7: Visual comparison with neural polygonizers. The result of PolyFormer (A1) and of Poly-RNN++ (B1) are simplified at different levels of complexity by Douglas-Peucker (A2-A3 and B2-B3 respectively). Precision, IoU and number of edges of the different results are given in the bottom graphs. At high and medium complexity, our method provides a more accurate approximation of the coyote (C1 and C2). For instance, only our method captures well the back of the animal or its two back legs. At low complexity, PolyRNN++ combined with Douglas-Peucker simplification have a better IoU, even if the precision of our method remains better.

Fig. 8: Visual results of object polygonalization (1/4). The ability of our line segments to capture both line-like structures and freeform shapes allows us to reconstruct low-complexity polygons with single edges on straight parts, e.g. on the racing cars, and nice smooth sequences of edges on curved parts, e.g. on the snake. Images from the *BSDS500* dataset.

Fig. 9: Visual results of object polygonalization (2/4).

Fig. 10: Visual results of object polygonalization (3/4).

Fig. 11: Visual results of object polygonalization (4/4). Note that the output polygons delivered by our method can be nested, ie with polygonal holes inside the main polygons. This is the case for instance with the handles of the amphora or in between the legs of the soldier.