

# Image partitioning into convex polygons

Liuyun DUAN      Florent LAFARGE  
INRIA Sophia Antipolis, France  
firstname.lastname@inria.fr

## Abstract

*The over-segmentation of images into atomic regions has become a standard and powerful tool in Vision. Traditional superpixel methods, that operate at the pixel level, cannot directly capture the geometric information disseminated into the images. We propose an alternative to these methods by operating at the level of geometric shapes. Our algorithm partitions images into convex polygons. It presents several interesting properties in terms of geometric guarantees, region compactness and scalability. The overall strategy consists in building a Voronoi diagram that conforms to preliminarily detected line-segments, before homogenizing the partition by spatial point process distributed over the image gradient. Our method is particularly adapted to images with strong geometric signatures, typically man-made objects and environments. We show the potential of our approach with experiments on large-scale images and comparisons with state-of-the-art superpixel methods.*

## 1. Introduction

The partitioning of images into meaningful atomic regions is very popular to address vision problems. When used as pre-processing for image segmentation [14], stereo matching [32] or object boundary extraction [12] for instance, such an image decomposition offers very interesting advantages in terms of algorithmic complexity and spatial consistency. Traditional methods create image partitions at the pixel level, atomic regions being commonly called *superpixels*. Each region is delimited by a set of pixels forming a free-form contour. This representation brings high flexibility, but is free of higher level information. In particular, it does not exploit geometric information disseminated into images, which can be a precious source of knowledge to analyze scenes and objects, especially in man-made environments.

In this paper, we address the problem of partitioning images into atomic regions with convex polygons while imposing geometric guarantees on the shape and connection of these regions. Figure 1 illustrates our goal.



Figure 1. Our algorithm partitions images into regular convex polygons. Three different polygon sizes are displayed. The use of floating polygons allow for the preservation of object boundaries at a subpixel scale (close-up).

### 1.1. Related works

Our review of previous work covers three main facets of our problem statement: segmentation into superpixels, shape detection, and object polygonalization.

**Segmentation into superpixels.** Methods partitioning images into superpixels are usually evaluated on five criteria: (i) adherence to boundaries, (ii) low running time, (iii) compactness of regions, (iv) memory efficiency and (v) simplicity of use. Among the numerous algorithms proposed in the literature, the most popular strategy consists in iteratively refining superpixels from an initial rough partitioning of pixels. These methods, eg [1, 13, 25, 28, 29], are usually time and memory efficient and capture well boundaries. Some methods address the problem with more global strategies, in particular with energy minimization on graph, eg [15, 20]. Results are usually of higher quality but require more algorithmic efforts. Globally speaking, each method has its own advantages and drawbacks, and scores differently on the five criteria mentioned above. Nevertheless, adherence to boundaries is usually favored at the expense of region compactness by a large majority of methods [21]. Apart from certain algorithms as SLIC [1], no control on the shape of regions is possible.

**Geometric shape detection.** The automated detection of geometric shapes is an instance of the general problem

of fitting parametric functions to data. There is a wide variety of shapes in all dimensions, the most common one in image problems being line-segments. This parametric shape is known to capture well the image discontinuities, in particular for man-made environments. Interpreting line-segments from images can bring precious information for discovering the scene structure [11] or recognizing people [18]. If the Hough detector has been widely used in the literature, recent algorithms deeply improved the quality of line-segment detection while guaranteeing fast running times [7], and even false detection control [27]. Closely related to line-segments, textons [31] also proposes a compact representation of the image structure in between the pixel and geometric shape scales.

**Object polygonalization.** Beyond simple geometric shapes, polygons also constitute valuable tools to capture objects or parts of objects. Generating polygons is usually performed by assembling line-segments preliminarily detected. In [9], convex polygons are extracted using a greedy search guided by local geometric constraints. Extracting free-form polygon is algorithmically more complex. It can be done, for instance, by searching for cycles in a graph of line-segments with a scoring function that measures the quality of cycles [22]. Another solution is to exploit a gap-filling strategy to connect the line-segments [30]. If polygons are common tools for capturing objects, their use is more marginal for interpreting entire scenes, or said differently, for partitioning images. One of the main reasons is the difficulty to create a partition of polygons that are perfectly connected, ie without overlap and empty space.

## 1.2. Positioning

Few works have addressed the problem of the geometric partitioning of images. Traditional superpixel decompositions are flexible and powerful tools for a large panel of computer vision applications, but cannot efficiently exploit the geometric knowledge disseminated into images. This is particularly penalizing in some applications or specific contexts for which the shape and adjacency of regions are expected to have strong geometric constraints. In stereo matching for instance, guaranteeing convexity of regions make the matching procedure more robust than the subsequent extraction of their convex hull [4]. Also, in presence of man-made objects and urban environments [19, 6], preferring regions with straight line boundaries is a natural choice which can be a precious source of geometric knowledge for subsequent processing steps.

Integrating such geometric knowledge after superpixel decomposition is a complex and delicate task. Inconsistencies within the graph of region adjacency are frequent and lead to generate structural incoherences in subsequent processing. Also, modifying the region shapes typically

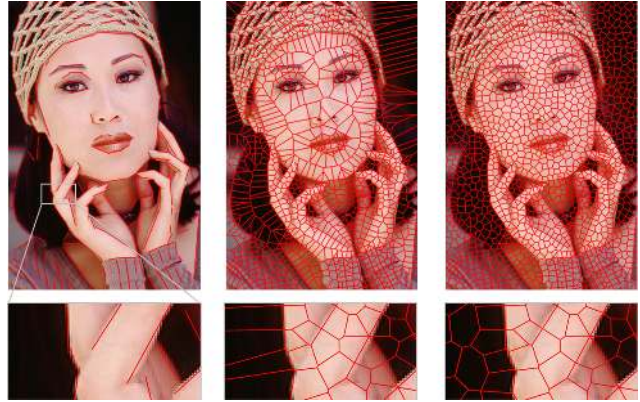


Figure 2. Overview. Left: line-segments are first extracted from the input image, and consolidated to bring spatial coherence (Sec.3). Middle: an initial Voronoi partition that preserves the line-segments and their junctions is then created by inserting anchors at some specific locations (Sec.4). Right: the Voronoi partition is homogenized by point process (Sec.5).

destroys the effort done to make superpixels adherent to the image. Ideally, both geometry and radiometry must be jointly exploited to generate the regions.

The proposed solution consists of partitioning images into connected convex polygons using Voronoi diagrams for which a brief introduction is given in Sec. 2. Region convexity has many advantages, in particular for (i) simplifying subsequent geometric operations as the computation of region distances, (ii) favoring the region compactness, and (iii) insuring a unique adjacency graph between regions, without ambiguities. In our approach, geometric properties are guaranteed by construction of the Voronoi diagram whereas radiometry is exploited to (i) align edges separating two neighboring polygons with image discontinuities, and (ii) center the polygons in homogeneous areas. Contrary to interest point based-strategies [24, 5], we approximate image discontinuities through the detection of geometric shapes, ie line-segments similarly to [22, 30].

## 1.3. Contributions

Our algorithm takes an image as input and produces, as output, a partition into polygons defined into the continuous bounded domain supporting the image. A model parameter  $\epsilon$  has to be specified to fix the partition scale; concretely  $\epsilon$  corresponds to the average radius of a region, assuming the region approaches a rounded shape. Our main contributions are as follows:

- *Shape anchoring.* We propose a strategy to preserve geometric shapes within the Voronoi partitions. The key idea relies on the insertion of pairs of Voronoi seeds, called *anchors*, close enough to each other to constrain the Voronoi edges to be part of a geometric shape. Beyond preservation, we also structure the

connexion of the geometric shapes, in particular for enhancing shape junctions.

- *geometric guarantees.* Our output provides some geometric guarantees related to the shape and adjacency of the atomic regions. First, each region is a convex polygon with a low number of edges. Contrary to many superpixel methods, the adjacency of regions is also guaranteed to be unique by construction, two polygons being neighbors if they share a common edge. Finally, region boundaries are polygons with exact geometry, ie under the pixel scale.
- *Efficiency.* By manipulating geometric entities, we simplify the pixel-based information and strongly reduce the algorithmic complexity of the partitioning process. If the efficiency of superpixel methods can be strongly affected by big size images, our algorithm is weakly impacted both in terms of time efficiency and memory consumption.

The proposed strategy is composed of three steps illustrated in Figure 2.

## 2. Mathematical background

We briefly introduce two mathematical tools that play a central role in our algorithm: Spatial point processes and Voronoi Diagrams. Deeper presentations of these tools can be found in [2, 17].

**Spatial point process.** A point process describes random configurations of points  $P = \{p_1, \dots, p_n\}$  in a continuous bounded set  $K$ , in our case the 2D domain of the input image. The number of points  $n$  is itself a random variable that typically follows a discrete Poisson distribution. What makes point processes appealing is the possibility to create spatial interactions between points, in particular using the Markovian property (points interact only in a local neighborhood). The most common process using Markovian interactions is the Strauss process in which a repulsion domain is located around each point to avoid points to be too close to each other. When  $\dim K = 2$ , this domain is a disk whose radius is a model parameter. The sampling of point process is usually a fastidious operation relying on Monte Carlo methods [26]. However, fast sampling mechanisms exist for certain types of point processes. This is the case of Strauss processes for which efficient Poisson-disk sampling allows the random generation of points either homogeneously distributed [8], or following an arbitrary density [3].

**Voronoi diagram.** Given a configuration of points  $P$  in  $K$ , called *seeds*, the Voronoi cell associated to the seed  $p_i \in P$ , denoted as  $V(p_i)$ , corresponds to the region in

which the points are closer to  $p_i$  than to any other seed in  $P$ :

$$V(p_i) = \{x \in K / \|x - p_i\| \leq \|x - p_j\|, \forall p_j \in P, i \neq j\} \quad (1)$$

The Voronoi diagram generated by  $P$  is the set of the Voronoi cells  $\{V(p_1), \dots, V(p_n)\}$ . Voronoi diagrams have interesting geometric properties, in particular they entirely partition the domain  $K$  without region overlap. By using the Euclidean distance in Eq. 1, Voronoi cells are guaranteed to be convex polygons. The dual graph of a Voronoi diagram also corresponds to the Delaunay triangulation of its seeds, and gives the adjacency relation between regions. Finally the algorithmic complexity to build a Voronoi diagram when  $\dim K = 2$  is only in  $O(n \log n)$ .

Spatial point processes can be used to generate the seeds of a Voronoi diagram. In particular, Poisson-disk sampling constitutes a fast and efficient way to create partitions of homogeneous Voronoi cells.

## 3. Shape detection

The first step of our algorithm consists of extracting line-segments from the input image, and then consolidating them to bring spatial coherence.

**Line-segment extraction.** As mentioned in Sec. 1.1, many methods have been proposed in the literature. Our choice focuses on the Line-Segment Detector (LSD) [27] for the detection quality, the running times and the false detection control. We fix the minimal length of line-segments to  $\epsilon$ . Note that our algorithm is not restricted to LSD and can be used with other line-segment or polyline extraction methods.

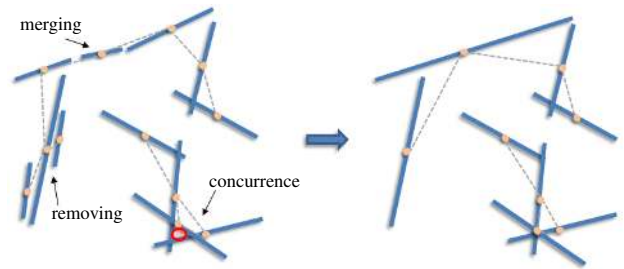


Figure 3. Line-segment consolidation. Three different operations applied greedily over the adjacency graph (dashed grey lines) bring spatial coherence between the detected line-segments. Such a consolidation procedure also reduces the problem complexity as the number of line-segments becomes lower.

**Consolidation.** The extraction of line-segments is a local process that can generate heap of shapes with noise and outliers. Such raw detected line-segments is sometimes hardly exploitable. We thus propose a consolidation procedure to bring spatial coherence between the line-segments.



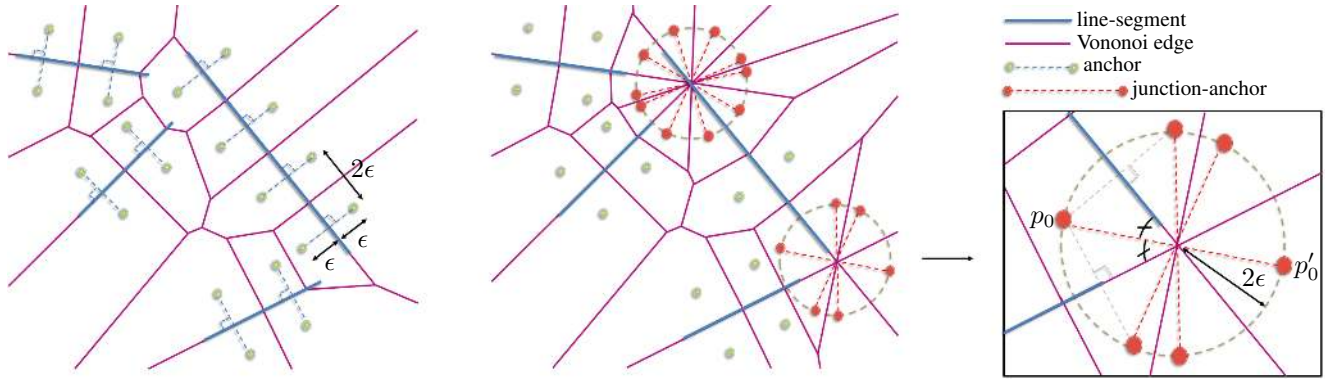


Figure 4. Anchoring. A set of anchors is positioned orthogonally to each line-segment, each seed of an anchor being at the same orthogonal distance  $\epsilon$  from the line-segment (left). Three (resp. five) junction-anchors are positioned to preserve junctions between two (resp. three) lines (middle). We start positioning the junction-anchor  $(p_0, p'_0)$  at the intersection of the junction-circle with the bisector of the line-segment pair having the smaller angle. We then create the other junction-anchors by orthogonal symmetry with respect to the line-segments (right).

An adjacency graph is built: two line-segments  $l_i$  and  $l_j$  are considered as adjacent if  $d(l_i, l_j) \leq \epsilon$ , where  $d(\cdot, \cdot)$  is the minimal euclidean distance between any pair of points of the two line-segments. As illustrated in Figure 3, we consolidate sets of adjacent line-segments using three types of operators:

- **Merging.** The merging operator tests whether two adjacent line-segments are near-collinear, and, if valid, replaces them by one large line-segment that covers their length.
- **Removing.** A small line-segment is removed when adjacent to a large near-parallel line-segment.
- **Concurrence.** The concurrence operator tests whether the inscribed circle of three mutually adjacent line-segments, ie of a simple cycle of order 3 in the adjacency graph, is small, and, if valid, translates the three line-segments onto the center of the inscribed circle.

Note the adjacency graph is updated after each effective operations. Merging, removing and concurrence operators are successively applied over the line-segments using a greedy procedure.

#### 4. Conforming Voronoi partition

Our objective is now to create a Voronoi partition that conforms to the detected line-segments. Said differently, the line-segments must not cross the Voronoi cells, but must be included onto the Voronoi edges. Manipulating a Voronoi partition to make the cells align with some geometric shapes or radiometric information is a delicate operation because, for the displacement of a single seed, even small, the whole group of connected cells is usually strongly perturbed. This explains why the use of Voronoi diagrams in vision has mainly been restricted to the creation

of basic isotropic partitions, eg in texture segmentation[23]. Inspired by a recent work in surface reconstruction to conform 3D Delaunay triangulation to planes [10], we propose a mechanism to create a Voronoi partition that conforms to the line-segments by construction.

**Shape anchoring.** The key idea consists in sampling pairs of seeds, that we call *anchors*, located on each side of a line-segment. As illustrated on Figure 4, each anchor is positioned so that the Voronoi edge separating the cells induced by the two seeds is exactly on the line-segment.

**Junction preservation.** The sampling of anchors is a local procedure on individual line-segments that does not preserve their junctions. We thus create *junction-anchors* by positioning pairs of seeds on a circle, called the *junction-circle*, centered at the intersection of the adjacent line-segments, and of radius  $2\epsilon$ . Anchors located inside junction-circles are first removed. Then, junction-anchors are inserted onto the junction-circle as explained in Figure 4. Note that three mutually adjacent line-segments are necessarily intersecting in one point as a consequence of the consolidation process.

The anchoring procedure is entirely controlled by the parameter  $\epsilon$ . Note that cells generated from junction-anchors have typically a triangular shape that reduce the global compactness of the partition. This is the price to pay for preserving the exact intersection of the line-segments into the partition. Junctions between at least four line-segments are marginal in practice: this case is not handled by our system.

#### 5. Spatial homogenization

The Voronoi partition from anchors generates cells of heterogeneous size. In particular, large cells poorly

captures the homogeneous areas of the input image. If line-segments capture well the main image discontinuities, they are less adapted to secondary boundaries, as those formed by the sail frames of the windmill on Figure 5. We thus refine the Voronoi partition by sampling a point process for a better spatial homogenization of polygons.

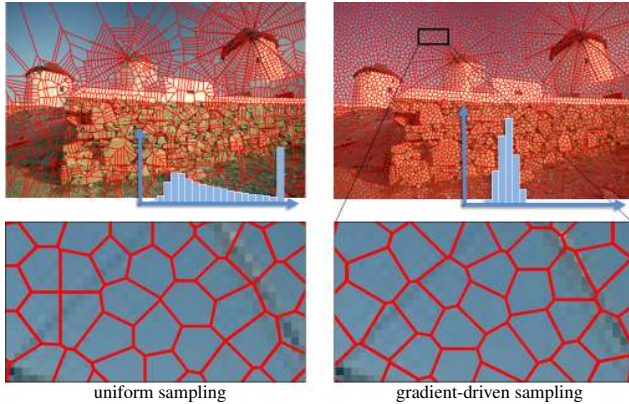


Figure 5. Spatial homogenization. Initial Voronoi partition from anchoring (top left) is refined into a partition (top right) with regular-sized cells (see histograms of the distribution of the Euclidean distance between boundary pixels and region centroid). When no line-segments are detected in an area, our Poisson-disk sampling driven by the image gradient allows the preservation of secondary boundaries contrary to a uniform sampling (see close-up).

**Sampling domain.** We first define a sampling domain so that the Voronoi edges supporting line-segments and their junction will not be affected by the insertion of new seeds. This domain is defined as the complementary, over the image domain, of the accumulated disks centered on each seed and of radius  $2\epsilon$ .

**Poisson-disk sampling.** New seeds are then distributed over this domain using a Poisson-disk sampling, the disk radius being equal to  $\epsilon$ . Instead of considering a homogeneous sampling, we guide the seed distribution with a spatial density, similarly to [3]. We define the density as proportional to the inverse of the image gradient, as detailed in Figure 6. The intuition behind that is to avoid new seeds to be positioned on image discontinuities. This procedure does not guarantee to produce Voronoi edges that perfectly align with secondary boundaries, but it encourages the positioning of seeds at the center of local homogeneous areas, as illustrated on Figure 5, bottom middle. Note that other types of spatial densities can also be used, eg texture or distance maps depending on the study context.

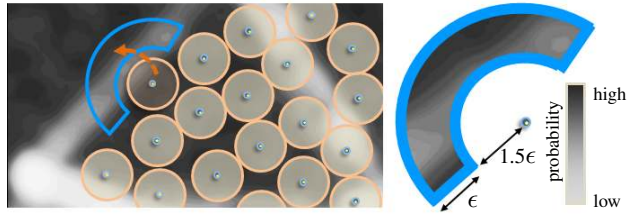


Figure 6. Poisson-disk sampling with non-homogeneous spatial distribution. Each new disk to insert is positioned into a circular domain of width  $\epsilon$  (blue contour). The sampling is guided by the inverse of the image gradient (grey scale), here from the close-up of Figure 5.

## 6. Experiments

The algorithm is implemented in C++, using the Computational Geometry Algorithms Library<sup>1</sup> for the Voronoi diagram structure as well as for the basic geometric operations as the computation of the line-segment distance. All timings are measured on an Intel Core i7 clocked at 2GHz. We experiment with both small size images from the Berkeley dataset and large size satellite images.

The main parameter of our algorithm,  $\epsilon$ , allows the control of the cell size. This parameter steps in the different stages our system. Four additional parameters are used during line-segment consolidation (Sec. 3): a maximal angle and a maximal distance to define the near parallelism and near-collinearity of line-segments, as well as a maximal radius of inscribed circle of three line-segments, and a minimal large to small line-segment length ratio. These four parameters are fixed respectively to  $5^\circ$ ,  $0.5\epsilon$ ,  $0.5\epsilon$  and 5 in all the experiments.

**Flexibility.** Because of the nature of the geometric shapes, our algorithm is particularly suitable for man-made environments in which boundaries are often accurately described by line-segments. It also produces convincing results on free-form boundary images as illustrated on Figures 2 and 9 (top row), even if the piecewise-linear approximation of object contours can be penalizing. Radiometric information are exploited at two different levels in the algorithm, ie during line-segment extraction and Poisson-disk sampling. The former plays a more important role as its conditions the positioning of the Voronoi edges onto the main image discontinuities.

**Comparison with superpixel methods.** Although our algorithm produces polygonal regions different from superpixels, it can be evaluated using the standard quality criteria required for superpixel methods. Four quality criteria are taken into account: boundary recall [1], undersegmentation error [13], compactness [21], and running times. We compare our algorithm on the Berkeley dataset [16] with three

<sup>1</sup>[www.cgal.org](http://www.cgal.org)

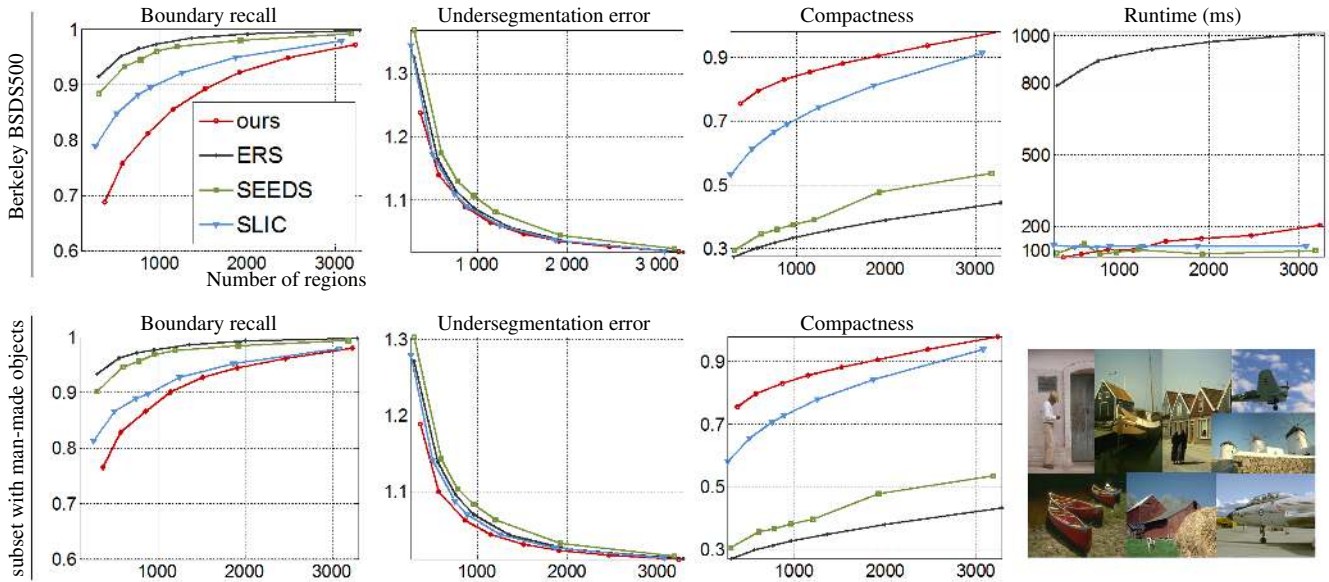


Figure 7. Quantitative evaluation on Berkeley dataset. Boundary recall, undersegmentation error, compactness and runtime are given for the entire dataset (top) and for a subset of 30 images in which man-made structures are dominant (see some samples in bottom right).

state-of-the-art superpixel methods: SLIC [1], SEEDS [25] and ERS [15]. For measuring the quality criteria on our method, the edges of the polygonal regions are discretized into pixel-based boundaries.

Figure 7 shows the results on the four quality criteria. Because our regions are convex polygons of homogeneous size, our algorithm outperforms the other methods in terms of compactness by a significant margin. The algorithm also competes well in terms of undersegmentation error and running time. Contrary to SEEDS and SLIC, our running time increases in function of the number of regions. Nevertheless, as we manipulate geometric objects, our algorithm is less impacted when the image size increases. In addition, our memory consumption is very low, even on very big images. Our result on the boundary recall scores low with respect to the three other methods. The use of line-segments logically penalizes the boundary accuracy, in particular when the number of regions is low. This is the price to pay to guarantee highly compact regions, boundary recall and compactness being hard to conciliate. Nevertheless, the boundary recall of our method improves when we restrict the evaluation to a subset of images for which man-made structures are dominant, as shown in Figure 7, bottom row. In particular, the boundary recall becomes quite close to SLIC. For such images, the boundary accuracy is less penalized by the use of line-segments. In terms of model parameters, our algorithm does not have a weight balancing between image faithfulness and region regularity as in SLIC or ERS. On the one hand, this characteristic reduces the flexibility of our algorithm. On the other hand, it allows us to guarantee some geometric properties (polygonal

shape, region convexity, unique adjacency graph) contrary to the other methods. Results with other quality criteria are presented in Supplementary Material.

Figure 9 shows some visual comparisons with these three superpixel methods. Our method competes well, specially for indoor and urban scenes. If the other methods typically perform better for capturing thin irregular details with large region size, we compensate by a higher region compactness, some geometric guarantees on the result, and region boundaries under the pixel scale.

**Large-scale satellite images.** Figure 8 shows a use case in which the algorithm characteristics are particularly attractive. Because of the scale and the geometric signature of the urban satellite images, the image partition preserves well shape of buildings, in particular the facade and rooftop edges, as well as building corners. This knowledge can be used later, for instance, in a geometry-aware classification of urban scenes.

|                 | church<br>154Kpixels | Manhattan<br>39.1Mpixels | Denvers<br>104Mpixels |
|-----------------|----------------------|--------------------------|-----------------------|
| line extraction | 36ms                 | 29.9s                    | 114.2s                |
| consolidation   | 3ms                  | 9.1s                     | 107.4s                |
| anchoring       | 3ms                  | 2.7s                     | 32.7s                 |
| homogenization  | 32ms                 | 10.2s                    | 48.4s                 |
| total time      | 72ms                 | 51.9s                    | 302.7s                |
| memory peak     | 12.63Mb              | 372.20Mb                 | 756.26Mb              |

Table 1. Performances on different image sizes (church from Figure 1, and Manhattan/Denvers from Figure 8) in terms of running time and memory consumption.



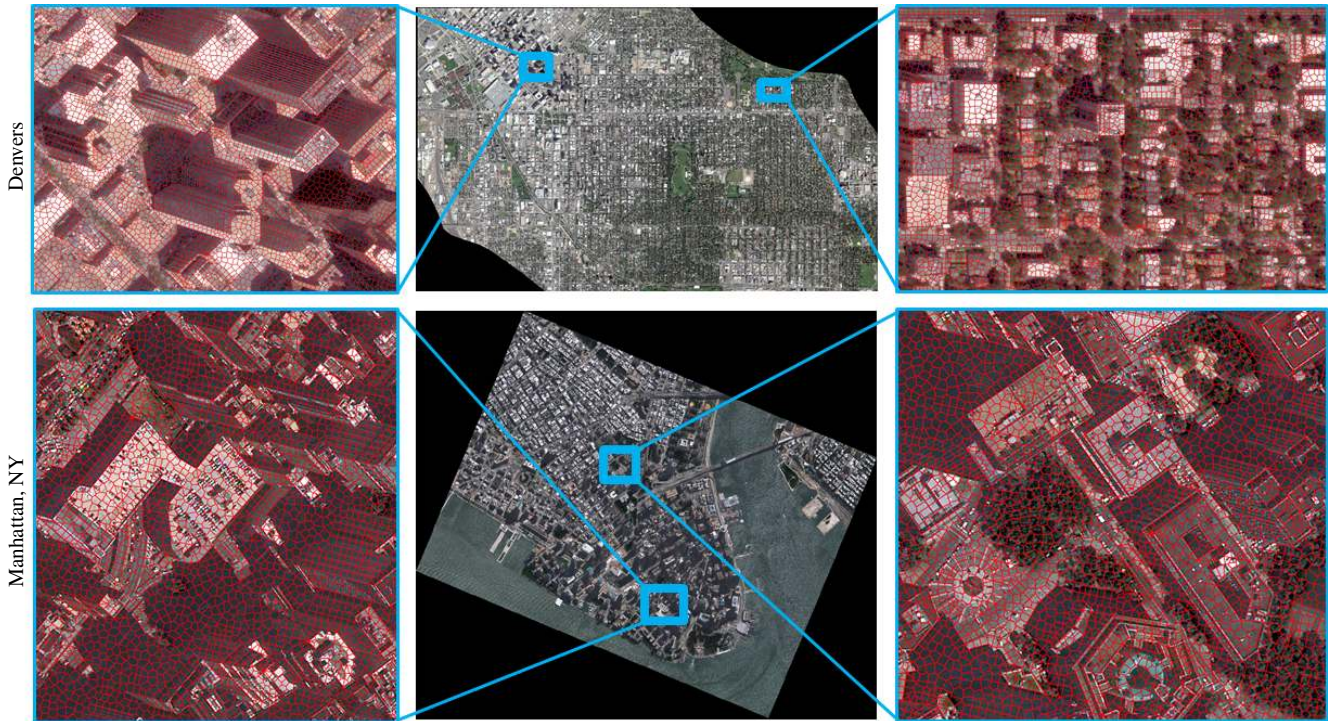


Figure 8. Large scale satellite images. Our algorithm decomposes a 104Mpixel (resp. 39Mpixel) image of Denver (resp. Manhattan) into 0.8M (resp. 0.2M) convex polygons in a few minutes. The image partition nicely preserves the facade edges and rooftop junctions in spite of low image contrast (see close-ups).

**Performances.** Our algorithm performs well from very big size images as shown in Table 1. Five minutes and 0.8Gb of memory are necessary from a 100Mpixel image. By contrast, the superpixel method ERS requires 39 minutes and 34Gb memory, and the released versions of SLIC and SEEDS do not run on such image size. Manipulating geometric shapes instead of pixels makes our algorithm particularly scalable. In terms of storage, our polygon partition can be saved in a very compact way as a planar graph where each node refers to a polygon.

**Limitations.** Our algorithm is designed to partition images with a polygonal approximation of region boundaries. If this approximation is usually relevant for man-made environments, it might be of lower interest for images with weaker geometric signatures. The accuracy of our results is also dependent of the quality of the detected line-segments. We used the state-of-the-art line-segment detector [27]: it produces accurate line-segments but still lacks of global regularization to get line-segment configuration of very high quality.

## 7. Conclusion

In this work, we propose a novel algorithm to partition images into convex polygons. Contrary to superpixel meth-

ods, we operate at the scale of the geometric shape, and not directly at the pixel scale. Our algorithm has demonstrated several interesting properties in terms of geometric guarantees, region compactness and scalability, and has shown potential for partitioning images with strong geometric signatures, typically man-made environments. The key technical ingredient of our work is an anchoring procedure to conform Voronoi diagrams to geometric shapes, more precisely to line-segments.

This work brings a geometric dimension to traditional superpixel segmentation methods. Used as preprocessing, we wish it will serve Vision to exploit more efficiently the geometric knowledge disseminated into images, for instance by polygonalizing objects with region grouping, classifying scenes at a subpixelic scale or matching regions for stereo. Some applications presented in Supplementary Material illustrate the potential of our approach in Vision.

The use of line-segments is however not fully adapted to images with weak geometric signatures. In future works, we would like to investigate the use of more flexible geometric shapes that capture better free-form objects. Quadrics or B-splines are potential solutions assuming we can build Voronoi diagram in non-Euclidean space that conform to these shapes.



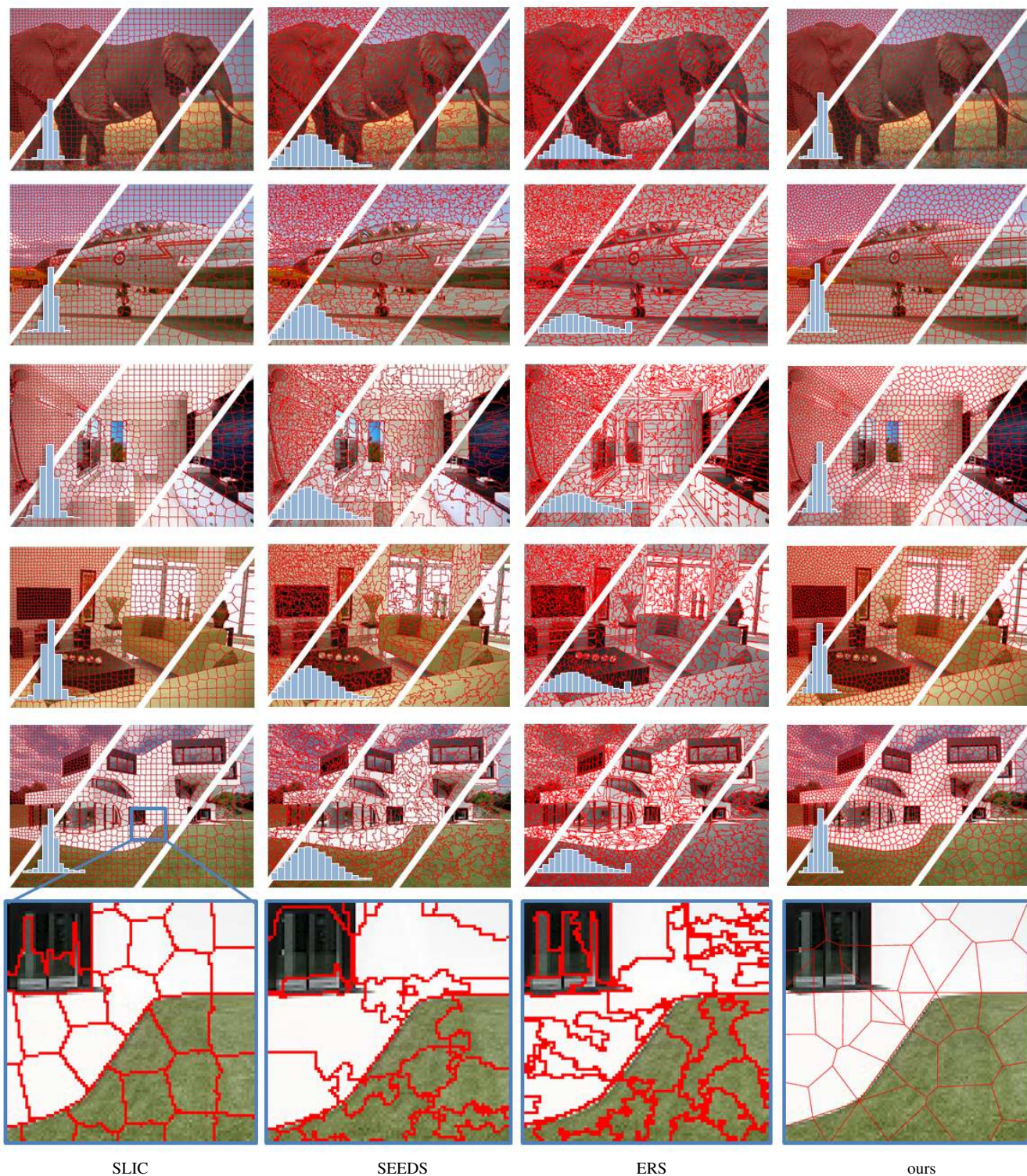


Figure 9. Visual comparison. Our algorithm produces competitive results for man-made objects or environments (four middle rows) in which the geometric structures are preserved. Only SLIC presents regions of the same order of compactness than ours algorithm, but with more outliers (see the histograms representing the distribution of the Euclidean distance between boundary pixels and region centroid, for the medium region size). Contrary to these methods, our regions are polygons able to preserve the geometric signatures of images at a subpixelic scale (see close-ups, bottom row).



## Acknowledgments

This work was supported by [Geoimage/Computamaps](#). The authors thank Lionel Laurore, Justin Hyland, Véronique Poujade and Frédéric Trastour for technical discussions and satellite datasets.

## References

- [1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *PAMI*, 34(11), 2012. [1](#), [5](#), [6](#)
- [2] A. Baddeley and M. Van Lieshout. Stochastic geometry models in high-level vision. *Journal of Applied Statistics*, 20(5-6), 1993. [3](#)
- [3] M. Balzer, T. Schlomer, and O. Deussen. Capacity-constrained point distributions: a variant of lloyd’s method. In *Proc. of Siggraph*, 2009. [3](#), [5](#)
- [4] A. Bodis-Szomoru, H. Riemenschneider, and L. Van Gool. Fast, approximate piecewise-planar modeling based on sparse structure-from-motion and superpixels. In *CVPR*, 2014. [2](#)
- [5] F. W. Chai, D. and F. Lafarge. Recovering line-networks in images by junction-point processes. In *CVPR*, 2013. [2](#)
- [6] G. Chaurasia, S. Duchene, O. Sorkine-Hornung, and G. Drettakis. Depth synthesis and local warps for plausible image-based navigation. *Trans. on Graphics*, 32(3), 2013. [2](#)
- [7] M. Dubska, A. Herout, and J. Havel. Pclines line detection using parallel coordinates. In *CVPR*, 2011. [2](#)
- [8] D. Dunbar and G. Humphreys. A spatial data structure for fast poisson-disk sample generation. In *Proc. of Siggraph*, 2006. [3](#)
- [9] D. Jacobs. Robust and efficient detection of salient convex groups. *PAMI*, 18(1), 1996. [2](#)
- [10] F. Lafarge and P. Alliez. Surface reconstruction through point set structuring. In *Proc. of Eurographics*, 2013. [4](#)
- [11] D. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, 2009. [2](#)
- [12] A. Levinstein, C. Sminchisescu, and S. Dickinson. Optimal contour closure by superpixel grouping. In *ECCV*, 2010. [1](#)
- [13] A. Levinstein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *PAMI*, 31(12), 2009. [1](#), [5](#)
- [14] Z. Li, W. Z.-M., and S.-F. Chang. Segmentation using superpixels: A bipartite graph partitioning approach. In *CVPR*, 2012. [1](#)
- [15] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy rate superpixel segmentation. In *CVPR*, 2011. [1](#), [6](#)
- [16] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001. [5](#)
- [17] A. Okabe, B. Boots, K. Sugihara, and S. Chiu. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley-Blackwell, 2000. [3](#)
- [18] X. Ren. Learning and matching line aspects for articulated objects. In *CVPR*, 2007. [2](#)
- [19] X. Ren, L. Bo, and D. Fox. RGB-(D) scene labeling: Features and algorithms. In *CVPR*, 2012. [2](#)
- [20] M. Reso, J. Jachalsy, B. Rosenhahn, and J. Ostermann. Temporally consistent superpixels. In *ICCV*, 2013. [1](#)
- [21] A. Schick, M. Fischer, and R. Stiefelhagen. Measuring and evaluating the compactness of superpixels. In *ICPR*, 2012. [1](#), [5](#)
- [22] X. Sun, M. Christoudias, and P. Fua. Free-shape polygonal object localization. In *ECCV*, 2014. [2](#)
- [23] M. Tuceryan, A. Jain, and Y. Lee. Texture segmentation using voronoi polygons. In *CVPR*, 1988. [4](#)
- [24] T. Tuytelaars. Dense interest points. In *CVPR*, 2010. [2](#)
- [25] M. Van den Bergh, X. Boix, G. Roig, B. De Capitani, and L. Van Gool. SEEDS: Superpixels extracted via energy-driven sampling. In *ECCV*, 2012. [1](#), [6](#)
- [26] Y. Verdier and F. Lafarge. Detecting parametric objects in large scenes by monte carlo sampling. *IJCV*, 106(1), 2014. [3](#)
- [27] R. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *PAMI*, 32(4), 2010. [2](#), [3](#), [7](#)
- [28] J. Wang and X. Wang. Vcells: Simple and efficient superpixels using edge-weighted centroidal voronoi tessellations. *PAMI*, 34(6), 2012. [1](#)
- [29] G. Zeng, P. Wang, J. Wang, R. Gan, and H. Zha. Structure-sensitive superpixels via geodesic distance. In *ICCV*, 2011. [1](#)
- [30] Z. Zhang, S. Fidler, J. Waggoner, Y. Cao, S. Dickinson, J. Siskind, and S. Wang. Superedge grouping for object localization by combining appearance and shape informations. In *CVPR*, 2012. [2](#)
- [31] S.-C. Zhu, C.-E. Guo, Y. Wang, and Z. Xu. What are textons? *IJCV*, 62(1/2), 2005. [2](#)
- [32] L. Zitnick and S. B. Kang. Stereo for image-based rendering using image over-segmentation. *IJCV*, 75(1), 2007. [1](#)