

Morphological Object Picking Based on the Color Tree of Shapes

Edwin Carlinet^{1,2} and Thierry Géraud¹

¹ EPITA Research and Development Lab. (LRDE)

14-16, rue Voltaire,

FR-94276 Le Kremlin-Bicêtre, France

{firstname.lastname}@lrde.epita.fr

² Université Paris-Est,

Laboratoire d'Informatique Gaspard-Monge (LIGM)

A3SI, ESIEE Paris, Cité Descartes, BP 99

FR-93162 Noisy-le-Grand, France

Abstract—The Tree of Shapes is a self-dual and contrast invariant morphological tree that provides a high-level hierarchical representation of images, suitable for many image processing tasks. Despite its powerfulness and its simplicity, it is still under-exploited in pattern recognition and computer vision. In this paper, we show that both interactive and automatic image segmentation can be achieved with some simple tree processings. To that aim, we rely on the “Color Tree of Shapes”, recently defined. We propose a method for interactive segmentation that does not involve any statistical learning, yet yielding results that compete with state-of-the-art approaches. We further extend this algorithm to unsupervised segmentation and give some results. Although they are preliminary, they highlight the potential of such an approach that works in the shape space.

Keywords—Mathematical Morphology; Tree of Shapes; Segmentation; Interactive segmentation.

I. INTRODUCTION

The Tree of Shapes (ToS) [1, 2] (also known as the level-line tree) has shown recently a regain of interest in the image processing community as it allows to represent efficiently the image as a hierarchical structure easy to manipulate. The abstraction offered by the Tree of Shapes enables to perform advanced image processing tasks in a simple way [3]. For example, image simplification can be performed by selecting or removing some nodes in the tree and image denoising can be computed by pruning some branches. Despite the simplicity of the framework, it has shown its powerfulness by competing with the state-of-the-art methods in many applications: the Tree-Based Morse Regions (TBMR) [4] achieve as well as the MSER for scene matching; the self-dual attribute profiles computed on the ToS get the state-of-the-art results for multispectral image classification [5], etc. The versatility of the structure is no more to demonstrate since it has been applied for visualization [6], filtering [7], object detection [8], image registration [9], and image simplification [10, 11]. It is not such a surprise if the Tree of Shapes achieves such good results but it is rather due to the mathematical properties held by the representation [12]. The Tree of Shapes is the support for self-dual, contrast invariant and morphological connected operators. In other words:

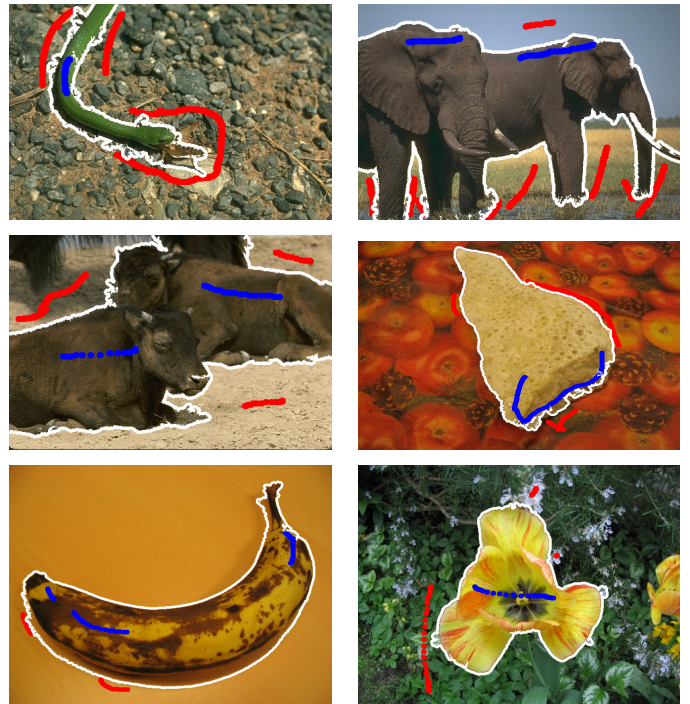


Fig. 1. Object picking with our method. Red and blue user scribbles define the background \mathcal{B} and the foreground \mathcal{F} respectively. The white line is the computed \mathcal{F}/\mathcal{B} boundary.

- 1) It enables to process dark objects over bright backgrounds and the contrary in a symmetric way.
- 2) It allows for recovering few contrasted objects.
- 3) It does not move the contours of the objects. Put differently, it features many properties of an image representation that one would like to have for image processing.

Yet, the Tree of Shapes is only defined for grayscale images as it requires a total ordering of the image values. We have shown that imposing a total order on colors leads to “strange” behaviors [13]. Following a former proposal [14], we have formalized in [15] a way to compute a Tree of Shapes on multivariate data that does not require to impose any total order. Just like the Tree of Shapes on scalar images, the Color

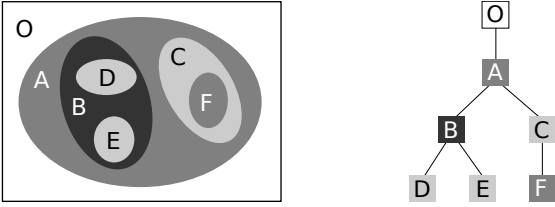


Fig. 2. An image (left) and its Tree of Shapes (right).

Tree of Shapes features many invariances on color images that are the invariances to any *marginal* change and/or inversion of contrast. More formally, let an image \mathbf{u} with values in \mathbb{R}^3 , $\mathbf{u} = (u_1, u_2, u_3)$, and let \mathbf{F} be the family $\mathbf{F} = (F_1, F_2, F_3)$ where $F_i : \mathbb{R} \rightarrow \mathbb{R}$ are monotonic functions:

$$\forall x, y \in \mathbb{R}, x < y \implies F_i(x) < F_i(y) \text{ or } F_i(x) > F_i(y).$$

Then, the Color Tree of Shapes $T(\mathbf{u})$ is invariant by marginal inversion/change of contrast that is $T(\mathbf{F}(\mathbf{u})) = T(\mathbf{u})$.

In this paper, we show the relevance of the Tree of Shapes for image segmentation, first in the context of interactive segmentation (see. Fig. 1) and then extending the method for unsupervised segmentation where the seeds are detected automatically without requiring any user intervention.

The paper is organized as follows. First, in Section II we remind the basics of the Tree of Shapes and its color extension and we present the mainlines of the segmentation method using the tree. Section III shows how our work differs from the other state-of-art object picking methods. Section IV gives an in-depth explanation of the proposed method, as well as a simple extension for non-supervised image segmentation. Section V shows some results using our proposal and we conclude in Section VI.

II. GENERAL OVERVIEW

A. The Color Tree of Shapes

Let an image $u : \Omega \rightarrow E$ defined on a domain Ω and taking values on a set E embedded with an ordering relation \leq . For any $\lambda \in \mathbb{R}$, a lower threshold set of u (also called lower cut of u) is defined by:

$$[u < \lambda] = \{x \in \Omega, u(x) < \lambda\}.$$

Similarly an upper threshold set of u (also called upper cut of u) is defined by:

$$[u > \lambda] = \{x \in \Omega, u(x) > \lambda\}.$$

We denote by $\mathcal{P}(E)$ the power set of E and by $\mathcal{CC}(X)$, with $X \in \mathcal{P}(E)$, the set of connected components of X . If \leq is a total relation, any two connected components $X, Y \in \mathcal{CC}([u < \lambda])$ are either disjoint or nested. The set $\mathcal{CC}([u < \lambda])$ endowed with the inclusion relation forms a tree called the *min-tree* and its dual tree, defined on upper cuts, is called the *max-tree*. Given the hole-filling operator \mathcal{H} , we call a *shape* any element of the set:

$$S = \{ \mathcal{H}(\Gamma), \Gamma \in \mathcal{CC}([u < \lambda]) \cup \mathcal{CC}([u > \lambda]) \}_{\lambda}.$$

If \leq is total, any two shapes are either disjoint or nested, hence the cover of (S, \subseteq) form a tree called the *Tree of Shapes* (see Fig. 2).

If \leq is not a total order (e.g., on colors), two shapes may overlap without being nested, and (S, \subseteq) does not form a tree anymore. Instead of defining a new total order, the Color Tree of Shapes [15] merges some Trees of Shapes computed marginally on each channel of the image into a single structure (a graph) based on the inclusion relation of connected components only. Then, a tree is extracted from the graph based on the graph topology and no more on values. This tree features many properties including the invariance to any marginal inversion/change of contrast.

B. Method Description

The problem of object picking can be summarized as follows: given two sets of points F and B in $\mathcal{P}(\Omega)$ representing the user inputs for the *foreground* class (\mathcal{F}) and the *background* (\mathcal{B}), we aim at classifying every point in Ω in one of the two classes.

Considering the distance measure between two points (p, p') in Ω :

$$d_{TV}(p, p') = \min_{C_{pp'}} \int_0^1 |\nabla u(C_{pp'}(t)) \cdot \dot{C}_{pp'}(t)| dt, \quad (1)$$

where $C_{pp'}(t)$ is a path in Ω from p to p' , we can compute the distance between any point x to the seeds F and B and assign the closest class to x . This approach has been used by [16, 17]. However, Dubrovina et al. [18] showed that the proper way to compute such a distance is to have a level set approach and to use the Tree of Shapes.

Thus, a fundamental idea of the method is to use the Color Tree of Shapes representation of the image (instead of working directly on the domain), and to perform the classification on that structure. It applies the same principle as previously (Nearest Neighbor) but uses the tree topology instead of the 2D space topology. The final segmentation is obtained by reconstructing the image from the tree where all nodes have been classified. The method can be summarized in the following steps (see also Fig. 3):

- 1) Compute the Color Tree of Shapes $T(\mathbf{u})$ of the image \mathbf{u} ,
- 2) Evaluate $T(\mathbf{u})$'s edges with the distance between nodes (see Section IV for more details),
- 3) Transpose the user scribbles on $T(\mathbf{u})$, it gives two seed sets of node for \mathcal{F} and \mathcal{B} ,
- 4) Classify every non-seed node as \mathcal{F} or \mathcal{B} by computing its distance to the seed nodes using $T(\mathbf{u})$'s topology, and retrieving the label of the closest seed node,
- 5) Reconstruct the image from the labels of $T(\mathbf{u})$,
- 6) Cleanup: keep significant foreground connected components only.

A significant advantage of working in the shape space is the ability to recover regions of interest that are not event marked by the user. This feature is interesting for objects composed by several other objects. Because a shape is a component without

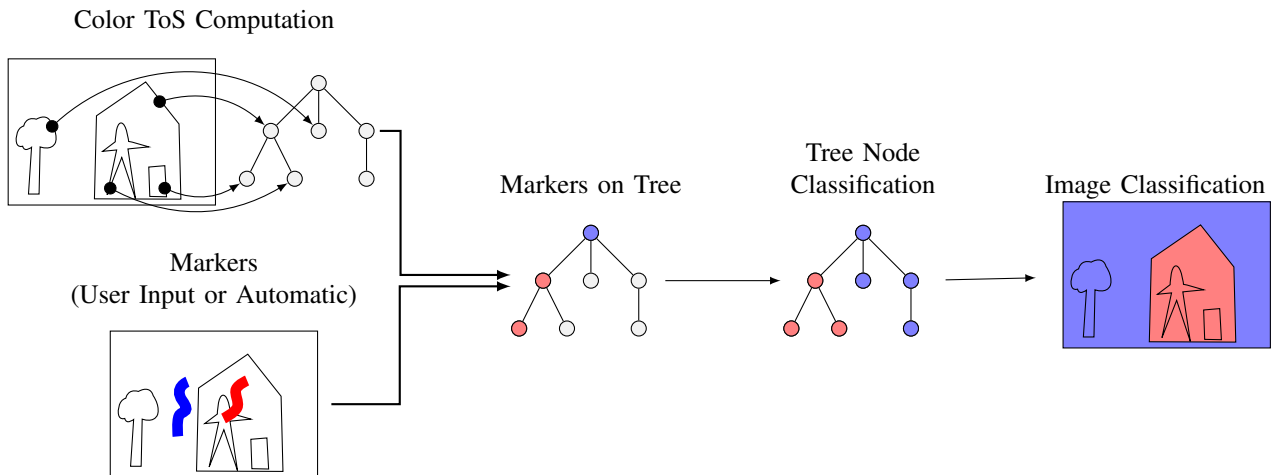


Fig. 3. Scheme of the proposed method for object picking.

holes, it is enough to select the outer region to retrieve the whole set of objects (see Fig. 4).

On the other hand, the approach does not compute any statistics about the regions, but only uses level sets that enable to recover large components with few user scribbles. The amount of markers required actually depends on the number of level lines that separate the background and the foreground.



Fig. 4. Object picking with and without holes. Because we are working in the shape space, tagging the outer object is enough to recover the whole region, but it does not prevent the user to get objects with holes if he wants to.

III. RELATED WORKS

Our approach is similar to the one proposed by Dubrovina et al. [18] since we both use the Tree of Shapes (or in our case the Color Tree of Shapes) to perform the classification of the nodes and then reconstruct the image from the labeled tree. The main difference is that Dubrovina et al. [18] compute the tree on the likelihood map where each pixel is the confidence to be a foreground pixel. It implies a statistical modeling of the user scribbles and so depends on the accuracy of the modeling of the probability function. Their work is actually an extension of [17] giving a better accuracy for the geodesic distance computation between unlabeled pixels and seeds. In their work, Bai and Sapiro [17] noted how important is an efficient density modeling method that enables a better segmentation while requiring less user scribbles than in their previous work [16]. But still, the quality of the estimation (so the results) highly depends on the trade-off between the region

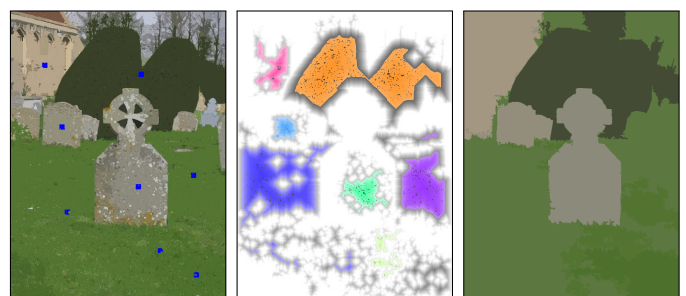


Fig. 5. Illustration of the automatic segmentation. Left: Image simplification with the α -tree ($\omega = 200$). Blue squares stand for the centers of the candidate regions ($\lambda = 3000$). Middle: The markers computed over the distance map. Right: The multi-classes segmentation based on the seeds found previously. The segmented components are shown with the average color over the region.

complexity and the number of user markers. Actually, most image editing or matting state-of-the-art algorithms, including grabcut [19, 20] (GMM modeling) use statistical learning for a background/foreground estimation, our method does not.

IV. ALGORITHM DETAILS AND EXTENSIONS

A. The ToS-based Object Picking Algorithm

The construction of the Color Tree of Shapes is out of the scope of the current paper, we only give a brief summary of it and we mainly detail in this section the way the ToS is used for segmenting (that are the steps 2–6 of Section II-B).

The first step involves the construction of a tree which features the same properties of the Tree of Shapes but for color images. To that aim, we have introduced in [15] the Color Tree of Shapes. Basically, it consists in computing the Tree of Shapes on each channel independently and merging all shapes in a single structure, called the Graph of Shapes, based on their inclusion. From this graph, a tree is computed through an intermediate depth image that represents the level of inclusion of each pixel in the set of shapes. More details can be found in [15].

The second step of the algorithm consists in evaluating a distance on tree edges. Contrary to the standard morphological trees, a node in the Color Tree of Shapes may be associated with many colors, as a consequence, we consider the average colors of the points in the node, and the edge between two nodes is evaluated with the distance between their average color (in $L^*a^*b^*$ space). Then, we need to label the nodes as \mathcal{F} or \mathcal{B} from the user scribbles. In some rare cases, a node may get contradictory labels if the user tags some points as \mathcal{F} and \mathcal{B} that belong to the same node. In that situation, the node gets labeled with the majority class.

The 4th step consists in computing for every non-labeled node S (shape) its distance $d_{\mathcal{F}}$ and $d_{\mathcal{B}}$ to the closest foreground and background seed. This can be achieved in two passes. We note $par(S)$, the parent of S in T , and $d(S, par(S))$ the distance between the average color of S and $par(S)$. At initialization, $d_{\mathcal{B}}(S) = 0$ if the node has background scribbles, $+\infty$ otherwise. The forward step goes from the leaves to the root, and computes:

$$d_{\mathcal{B}}(par(S)) = \min(d_{\mathcal{B}}(par(S)), d_{\mathcal{B}}(S) + d(S, par(S))).$$

The backward step goes from the root to the leaves, and computes:

$$d_{\mathcal{B}}(S) = \min(d_{\mathcal{B}}(S), d_{\mathcal{B}}(par(S)) + d(S, par(S))).$$

The same computation process holds for $d_{\mathcal{F}}$ and finally, a node gets labeled with $\arg \min_{C \in \{\mathcal{F}, \mathcal{B}\}} d_C(S)$.

In the tree $T(\mathbf{u})$, every point belongs to a single node, so the reconstruction of the image consists in assigning to each point the label of the node it belongs to. Eventually, the clean-up step consists in removing non significant regions that are connected components whose size is below half the size of the largest connected component.

The overall complexity of the method is quasi-linear in the number of points, which is the complexity of computing the Color Tree of Shapes [21, 22].

B. Extension for Automatic Segmentation

The previous framework requires the user to input the scribbles for tagging foreground and background objects. The objective is now to extend this algorithm for automatic segmentation without any user intervention. Instead of a binary segmentation, we first extend the algorithm to a multi-classes classification without any effort. Given a set of marked points $\{M_1, \dots, M_n\}$ labeled respectively with classes C_1, \dots, C_n , we still aim at classifying any other non-labeled data in Ω . We still use the Nearest Neighbor approach using the distance defined by Eq. (1). The binary to multi-classes classification is thus straightforward. The real challenge is to get automatically the initial set of seeds. For this purpose, we can rely on any initialization method used by other segmentation algorithms (e.g. the minima of the gradient function as for the Watershed Transform, the modes of the density function as for the Mean Shift algorithm...). We arbitrary have chosen to initialize our algorithm from the (α, ω) -flat zones of the image that are the set of connected components Γ in the image where the

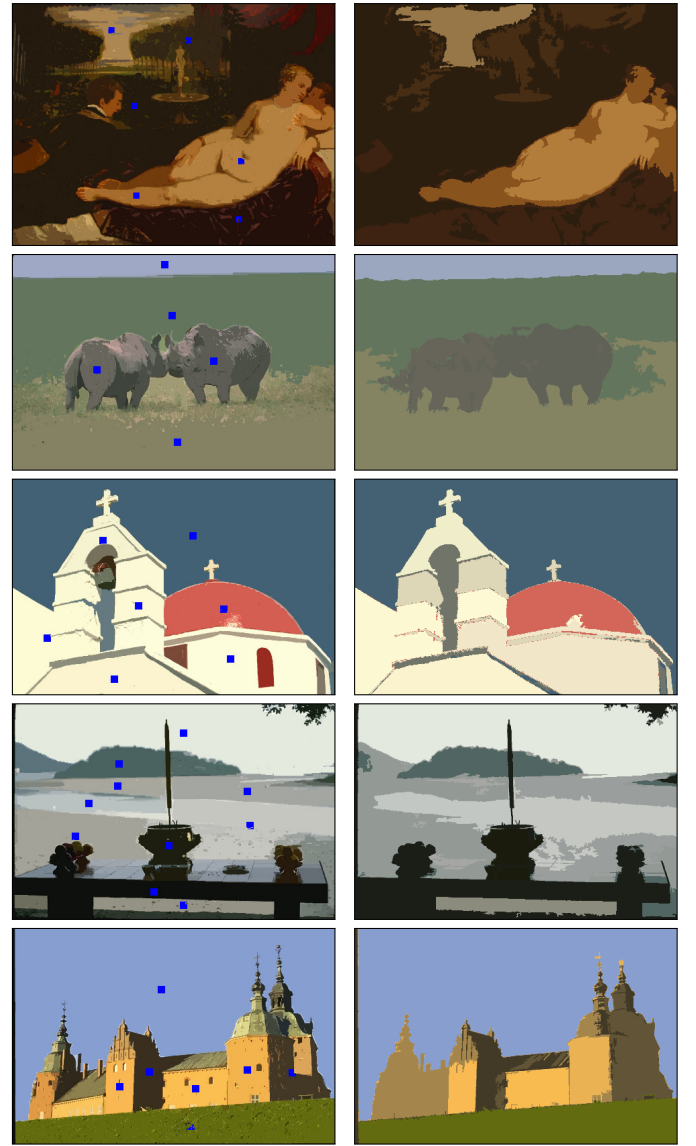


Fig. 6. Example of automatic segmentation on the BSD dataset [23]. Left: original images, where blue squares are the markers computed from the α -tree; Right: results of our segmentation method.

magnitude of the gradient inside Γ cannot exceed α and the amplitude of the range cannot exceed ω [24]. More formally, the initial set of seeds comes from:

$$\{\Gamma_i\} = \left\{ \Gamma \in \bigcup_{\alpha} CC_{\alpha}(u) \mid \max(u(\Gamma)) - \min(u(\Gamma)) < \omega \right\}$$

where $CC_{\alpha}(u)$ denotes the set of α -connected components of \mathbf{u} . The set $\{\Gamma_i\}$ forms a partition of the image but leads to an over-segmentation (especially on object boundaries with high gradients), thus we only keep the subset $\{\Gamma'_i\}$ of $\{\Gamma_i\}$ whose size is above a given threshold. Finally, we compute a thick skeleton of the candidate regions that will be the final markers. The process can be summarized as follows (see also Fig. 5):

- 1) Compute the α -tree of \mathbf{u} and retrieve the largest α -connected components such that their range does not

- exceed w and their size is above a given threshold λ .
- 2) For each component, compute a distance map from the background and threshold this distance map. It yields thick skeletons of the components that are the automatic markers.
 - 3) Perform the multi-classes segmentation extension of the algorithm seen in Section IV.

V. EXPERIMENTS AND DISCUSSION

Figures 1 and 6 show some results of the interactive segmentation and automatic segmentation using our approach (implemented using our platform Olena [25, 26]). At this point, and despite its simplicity, our approach for interactive segmentation competes with the state-of-the-art methods such as Grabcut from a qualitative point of view. Due to limited space, a qualitative comparison of our method with some other ones is available in [27], as well as more details about the experimental procedure. However it is still premature to have a quantitative assessment with those approaches as we have to tackle some problems first. A common failure case of our method appears when the object of interest is traversed by some level-lines of the background. In that case, we cannot separate correctly the foreground and the background. We have observed in recent experiments that this problem can be partially solved by a different rooting of the Color Tree of Shapes (by choosing the point at infinity in the foreground or background scribbles when constructing the tree). The automatic segmentation extension of this work is still in development and is outperformed by the current state-of-art segmentation methods. The method widely depends on the way the seeds are chosen and those given by the α -tree mainly appear in the background. We have to first improve the seed computation process before any further comparison with other methods. However, regarding the quality of the current results, it tends to show the potential of a direct classification in the shape space and that the Color Tree of Shapes is an adequate structure for segmentation.

VI. CONCLUSION

We have shown the versatility and the potential of the Tree of Shapes for segmentation purposes. We have proposed a marker-based classification using the Color Tree of Shapes which is free of statistical learning and despite its simplicity, yields results similar to the ones obtained by the state-of-the-art methods. We have also presented an extension that enables an automatic segmentation without requiring any user intervention. The automatic segmentation, still at a development process, provides interesting results that show the potential of the Color Tree of Shapes and the advantages of working in the shape space. As a further work, since our approach is already robust without statistical learning, we plan to introduce more sophisticated classification strategies to improve the interactive segmentation that would enable to further compare our work with other state-of-the-art methods. For the unsupervised segmentation part of this work, since the bottleneck lies in the automatic detection of the seeds, we

plan to experiment other pre-processing methods to get better markers.

REFERENCES

- [1] V. Caselles, B. Coll, and J.-M. Morel, "Topographic maps and local contrast changes in natural images," *International Journal of Computer Vision*, vol. 33, no. 1, pp. 5–27, 1999.
- [2] P. Monasse and F. Guichard, "Fast computation of a contrast-invariant image representation," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 860–872, 2000.
- [3] E. Carlinet and T. Géraud, "A comparative review of component tree computation algorithms," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 3885–3895, September 2014.
- [4] Y. Xu, P. Monasse, T. Géraud, and L. Najman, "Tree-based morse regions: A topological approach to local feature detection," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5612–5625, 2014.
- [5] G. Cavallaro, M. D. Mura, J. A. Benediktsson, and L. Bruzzone, "A comparison of self-dual attribute profiles based on different filter rules for classification," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2014, pp. 1265–1268.
- [6] Y. Xu, E. Carlinet, T. Géraud, and L. Najman, "Meaningful disjoint level lines selection," in *Proceedings of the 21st IEEE International Conference on Image Processing (ICIP)*, Paris, France, 2014, pp. 2938–2942.
- [7] Y. Xu, T. Géraud, and L. Najman, "Two applications of shape-based morphology: Blood vessels segmentation and a generalization of constrained connectivity," in *Mathematical Morphology and Its Applications to Signal and Image Processing—Proceedings of International Symposium on Mathematical Morphology (ISMM)*, ser. Lecture Notes in Computer Science, C. L. Hendriks, G. Borgefors, and R. Strand, Eds., vol. 7883. Springer, 2013, pp. 390–401.
- [8] —, "Context-based energy estimator: Application to object segmentation on the tree of shapes," in *Proceedings of the 19th IEEE International Conference on Image Processing (ICIP)*, Orlando, Florida, USA, October 2012, pp. 1577–1580.
- [9] V. Caselles and P. Monasse, *Geometric Description of Images as Topographic Maps*, ser. Lecture Notes in Mathematics. Springer-Verlag, 2009, vol. 1984.
- [10] C. Ballester, V. Caselles, L. Igual, and L. Garrido, "Level lines selection with variational models for segmentation and encoding," *Journal of Mathematic Imaging and Vision*, vol. 27, pp. 5–27, 2007.
- [11] Y. Xu, T. Géraud, and L. Najman, "Salient level lines selection using the Mumford-Shah functional," in *Pro-*

- ceedings of the *IEEE International Conference on Image Processing (ICIP)*, 2013, pp. 1227–1231.
- [12] F. Cao, J.-L. Lisani, J.-M. Morel, P. Musé, and F. Sur, *A Theory of Shape Identification*, ser. Lecture Notes in Mathematics. Springer, 2008, vol. 1948.
- [13] E. Carlinet and T. Géraud, “Getting a morphological tree of shapes for multivariate images: Leads, traps, and pitfalls,” in *Proceedings of the 21st IEEE International Conference on Image Processing (ICIP)*, Paris, France, 2014, pp. 615–619.
- [14] —, “A morphological tree of shapes for color images,” in *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR)*, Stockholm, Sweden, August 2014, pp. 1132–1137.
- [15] —, “A color tree of shapes with illustrations on filtering, simplification, and segmentation,” in *Mathematical Morphology and Its Applications to Signal and Image Processing—Proceedings of International Symposium on Mathematical Morphology (ISMM)*, ser. Lecture Notes in Computer Science, J. Benediktsson, J. Chanussot, L. Najman, and H. Talbot, Eds., vol. 9082. Reykjavik, Iceland: Springer, 2015, pp. 363–374.
- [16] A. Protiere and G. Sapiro, “Interactive image segmentation via adaptive weighted distances,” *IEEE Transactions on Image Processing*, vol. 16, no. 4, pp. 1046–1057, 2007.
- [17] X. Bai and G. Sapiro, “A geodesic framework for fast interactive image and video segmentation and matting,” in *Proceedings of International Conference on Computer Vision (ICCV)*, 2007, pp. 1–8.
- [18] A. Dubrovina, R. Hershkovitz, and R. Kimmel, “Image editing using level set trees,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Paris, France, Oct. 2014, pp. 4442–4446.
- [19] C. Rother, V. Kolmogorov, and A. Blake, “GrabCut: Interactive foreground extraction using iterated graph cuts,” *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, Aug. 2004.
- [20] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp, “Image segmentation with a bounding box prior,” in *Proceedings of International Conference on Computer Vision (ICCV)*. IEEE, 2009, pp. 277–284.
- [21] T. Géraud, E. Carlinet, S. Crozet, and L. Najman, “A quasi-linear algorithm to compute the tree of shapes of n -D images,” in *Mathematical Morphology and Its Applications to Signal and Image Processing—Proceedings of International Symposium on Mathematical Morphology (ISMM)*, ser. Lecture Notes in Computer Science, C. L. Hendriks, G. Borgefors, and R. Strand, Eds., vol. 7883. Uppsala, Sweden: Springer, 2013, pp. 98–110.
- [22] S. Crozet and T. Géraud, “A first parallel algorithm to compute the morphological tree of shapes of n D images,” in *Proceedings of the 21st IEEE International Conference on Image Processing (ICIP)*, Paris, France, 2014, pp. 2933–2937.
- [23] 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 *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV)*, vol. 2, 2001, pp. 416–423.
- [24] P. Soille, “Constrained connectivity for hierarchical image partitioning and simplification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1132–1145, 2008.
- [25] R. Levillain, T. Géraud, and L. Najman, “Milena: Write generic morphological algorithms once, run on many kinds of images,” in *Mathematical Morphology and Its Applications to Signal and Image Processing—Proceedings of International Symposium on Mathematical Morphology (ISMM)*, ser. Lecture Notes in Computer Science, M. H. F. Wilkinson and J. B. T. M. Roerdink, Eds., vol. 5720. Springer, August 2009, pp. 295–306.
- [26] —, “Why and how to design a generic and efficient image processing framework: The case of the Milena library,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Hong Kong, September 2010, pp. 1941–1944.
- [27] E. Carlinet, “A tree of shapes for multivariate images: Definition and applications,” Ph.D. dissertation, Université Paris-Est, France, 2015, to appear.