A Morphological Tree of Shapes for Color Images

Edwin Carlinet*†

*Université Paris-Est,
Laboratoire d'Informatique Gaspard-Monge (LIGM),
A3SI, ESIEE Paris, Cité Descartes, BP 99
FR-93162 Noisy-le-Grand, France
edwin.carlinet@lrde.epita.fr

Thierry Géraud †

†EPITA Research and Development Laboratory
(LRDE)
14-16, rue Voltaire
FR-94276 Le Kremlin-Bicêtre, France
thierry.geraud@lrde.epita.fr

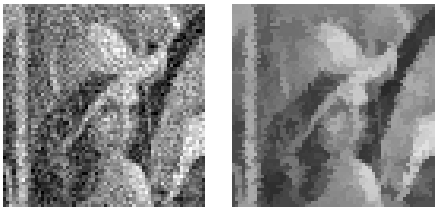
***Abstract*—In mathematical morphology the tree of shapes of a gray level image is a versatile representation that allows for multiple powerful applications. That structure is highly interesting because it is a self-dual representation invariant by contrast changes and since many authors state that object contours are well described by level lines. Such a representation has not yet been defined (thus used) on color images because *a priori* a total order on colors is required that really make sense on data. In this paper we propose a solution to obtain a tree of shapes on color images without resorting to an ordering of colors. To that aim we relax the definition of shapes and we show that relevant applications follow from our proposal.**

***Keywords*—Mathematical morphology, color images, tree of shapes, connected filters.**

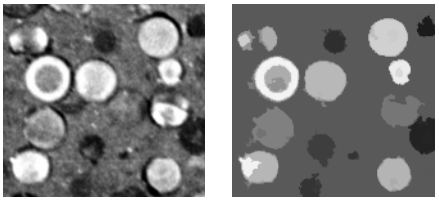
I. Introduction

Mathematical morphology operators can be divided into two large classes: the most known operators make use of structuring elements, whereas connected operators [1] are based on neighborhoods and connected components. The proeminent property of the latter is that they do not shift contours. Many connected filters on gray level images are dual, for instance, algebraic openings and closings [2], and can be defined from the min-tree and/or the max-tree. This couple of dual trees represent the image and encodes that connected components of respectively lower and upper thresholds cuts (also called *cuts* [3]) form a tree w.r.t. component inclusion. A self-dual tree has been defined by Monasse and Guichard [4], called *tree of shapes*.It describes the image contents in a unique way; such a tree can be understood as the result of merging the dual trees components. Some new connected operators can be derived from that tree, that make no assumption about the contrast of image components: the inclusion relationship can be due either to light objects surrounded by darker ones, or to the contrary. As a consequence self-dual operators process light and dark objects in the same way.

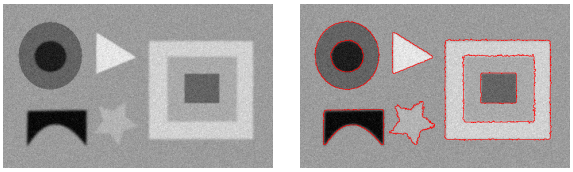
The tree of shapes is a morphological tool with a strong potential, that has been underexploited. Though its applications are numerous: image filtering [5, 6], simplification [7], and segmentation [8, 9, 10], and also texture indexing [11] and object recognition [12] (additional references about applications can be found in [13]). The reason why the tree of shapes is interesting is reported by several authors who claim that, in gray level images, object contours coincide with level lines [14, 15, 16]. Indeed, in [17] we have shown that object detection based on the tree of shapes can outperform the Chan and Vese method.



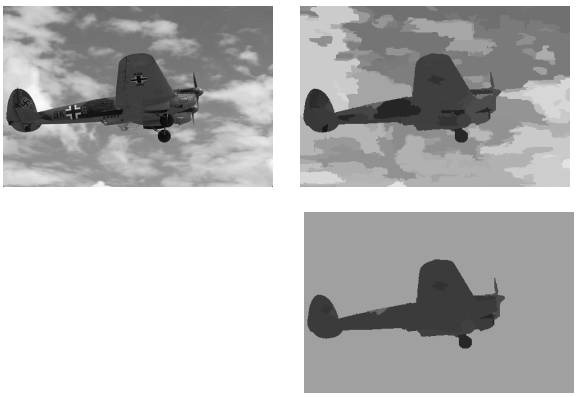
(a) Denoising (self-dual grain removal).



(b) Shape Filtering (keeping round objects) [6].



(c) Object Detection (energy-based method) [17].



(d) Hierarchical Segmentation (top: fine; bottom: coarse) [18].

Fig. 1. Sample uses of gray-level tree of shapes (left column: input images; right column: state-of-the-art results).

Yet obtaining a tree of shapes for color images is challenging. In a gray level image f, shape contours are level lines; they thus correspond to connected components of $\{x \mid f(x) = \lambda\}$, where λ is a gray level. Transposing that definition to a color image means that we expect every objects contained in this image to be surrounded by a curve which has a constant color λ. Of course that hardly happens in images, which makes this definition of shapes unusable in practice. Another idea to construct a tree of color shapes is to rely on the classical approaches of adapting a gray-level-based method or operator to color images, as performed in [10] and [19]. Unfortunately we will see that the trees obtained that way are not so well formed since *a posteriori* applications (noise removal and segmentation) do not lead to proper results.

To the authors knowledge the present paper is the first attempt to define a tree of shapes for color images *without* imposing some arbitrary or questionable ordering of colors. This paper is organized as follows. In section II we take a tour of the classical approaches for processing color images that can be used to construct trees of shapes. In section III we detail our proposal and we present, compare and discuss the results in section IV. Finally in section V we conclude and give some perspectives of our work.

II. State of the Art

When dealing with hierarchical representations of images, we can distinguish two classes of trees: hierarchies of segmentation and morphological trees. The first approach is directly linked to hierarchical clustering. It aims at either growing and merging regions in a bottom-up fashion or splitting regions in a top-down fashion. Among some well-known hierarchical segmentation trees [20], ultrametric wathersheds [21], binary partition trees [22], or alpha-trees [23] have been successfully applied to segmentation, object detection, image matching [6, 24]... The force of those methods when dealing with colors lies in that they only require a *distance* function between values. Any norm on colors has sense (even if some colorspace or norms may be closer from human perception than others), whereas an ordering of the colorspace is less obvious. Unfortunately, a total ordering is required to compute the second class of trees. While hierarchical segmentation trees consider clustering of adjacent region, morphological trees focus on the inclusion relationship of components. The components result from thresholding the image at different levels, and a total order on the image levels ensure the inclusion of the level sets. In what follows, we present the classical approaches to deal with the absence of a natural total order.

A. Marginal Processing

A common and simple approach when dealing with multivariate data is split each component and process them independently. In our case, it would result in computing trees of shapes for the red, green, and blue channels of the original image and filter each of them. Red, blue and green images are reconstructed from the three trees of shapes and merged back to give the final result. Although this simple method gives visually good results when the objective remains filtering or image simplification, it suffers from two majors drawbacks. First, algebraic properties of grain filters are not preserved. Filtering red, blue and green images with a grain of size λ implies that extreme components have an area greater than λ. However, since these components do not share the same location, the merge procedure introduces component overlap and noise in large flat zones. Second, this method remains useless for computer vision purposes. Since it does not build a single hierarchical representation of the image but rather three trees, computer vision methods that relies on node identification in a single tree cannot be applied.

B. Imposing a Total (Pre-)Order

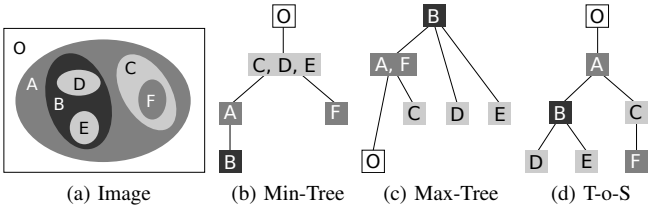
A classification of ordering relations can rely on their algebraic properties (totality, anti-symmetry...) or with respect to the way these relations are built. Barnett [25] proposed to classify them into four groups: marginal ordering (M-ordering), conditional ordering (C-ordering), partial ordering (P-ordering) and reduced ordering (R-ordering). M-ordering, is a component-wise ordering that deals separately with each channel. Since it is a partial order, it is out of scope of the current section.

In C-ordering, vectors are ordered by mean of one, several, or all of their marginal components. The most well-known C-ordering is the lexicographical ordering, that is a total order. If only some components participate in the comparison, it yields to a total-pre-order. For example let $v, w \in \mathbb{R}^n$, the lexicographical ordering $\leq$ using only the first two components defined as $v \leq_L w$ iff. $(v_1 < w_1) \vee (v_1 = w_1 \wedge v_2 \leq w_2)$ is a total-preorder. Colors $(1, 1, 2)$ and $(1, 1, 3)$ are considered as *equivalent* by the above relation. The main pitfall of the C-ordering resides in the importance given to first components. In the case of the RGB space, it implies for example that the red component is more relevant than the others. Several workarounds have been proposed [26, 27] like changing the color space (LSH is commonly used) or sub-quantization of first components (known as α-lexicographical ordering). This enables to lower the importance given to the first dimension but still introduce some color artifacts.

In R-ordering, vectors are reduced to scalar values using a mapping $h : \mathbb{R}^n \to \mathbb{R}$. Then, for two vectors $v, w \in \mathbb{R}^n$ a new relation $\leq_R$ is defined as $v \leq_R w$ iff. $h(v) \leq h(w)$. If h is injective then each index is mapped to a unique color and the relation is a total order, otherwise it is a total pre-order. Typical examples of R-ordering are distance-based orderings [28]. It consists in choosing a reference vector (or a reference set of vectors) v_{ref} and the order relation is built upon a distance to v_{ref}, i.e., $v \leq_R w$ iff $d(v, v_{ref}) \leq d(w, v_{ref})$. The main drawback of distance-based orders lies in the choice of a reference vector (or a set of reference vectors).More sophisticated methods can use statistical learning to rank colors [29, 30] but actually they are yet another kind of R-ordering.

C. Other Approaches

Since the natural order on colors is partial and forms a lattice, Passat and Naegel [31] proposed an extension of min trees (resp. max-trees) to lattice representing inclusion of lower cuts (resp. upper cuts). Since this representation is not self-dual, it is out of scope of this topic and the study of a *self-dual lattice of shapes* is postponed as a further work.



(a) Image (b) Min-Tree (c) Max-Tree (d) T-o-S

Fig. 2. An image (a), and its morphological component trees (b) to (d).

III. Proposed Method

A. Basic Notions

Let an image $f : \Omega \to (E, \leq)$ and $\lambda \in E$. We note $[f \leq \lambda]$ (resp. $[f \geq \lambda]$) a lower cut (resp. upper cut) of f defined as $[f \leq \lambda] = \{x,\ f(x) \leq \lambda\}$. Let $CC(X), X \in \mathcal{P}(E)$, denote the set of connected components of X. If $\leq$ is a total relation, any two connected components $X, Y \in CC([f < \lambda])$ are either disjoint or nested, thus the set $CC([f < \lambda])$ endowed with the inclusion relation forms a tree called the *min-tree* (see Figure 2b). Its dual tree, defined on upper cuts, is called the *max-tree* (see Figure 2c).

Let the saturation operator $Sat : \mathcal{P}(E) \to \mathcal{P}(E)$, defined as $Sat(X) = X \cup Y$ with Y *hole* of X, s.t. $Y \in CC((\Omega \backslash x)^c), Y \subset X$. We call a *shape* any element of $Sat([f \leq \lambda]) \cup Sat([f \geq \lambda])$. If $\leq$ is total, any two shapes are either disjoints or nested, hence form a tree called the *tree of shapes* (see Figure 2d).

B. Method Description

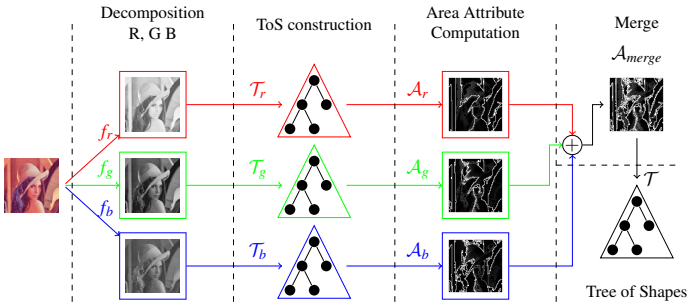


Fig. 3. Scheme of the proposed method.

The method relies on the simple observations that marginal filtering described in the previous section gives visually good results while methods introducing a total order introduce color artifacts as well. As, we require a unique hierarchical representation, we aim at merging the three trees of shapes from marginal processing in a single tree $\mathcal{T}$ while preserving the following three algebraic properties of trees of shapes:

1) a node is a connected component without holes (a shape);
2) removing a node from $\mathcal{T}$ boils down to merging some f flat zones;
3) parent relationship of nodes reflects shape inclusion in f.

The proposed method consists in the following five steps, also depicted in Figure 3:

Step 1. Decompose the color image into its three (red, green, and blue) channels images.
Step 2. Independently compute their respective trees of shapes $(\mathcal{T}_r, \mathcal{T}_g, \mathcal{T}_b)$.
Step 3. Compute area attributes as images $(\mathcal{A}_r, \mathcal{A}_g, \mathcal{A}_b)$. To that aim, each node is valuated with its area, e.g., the number of pixels of the shape it represents; attribute images are reconstructed from trees, with each pixel getting its value from the corresponding node's area.
Step 4. Merge attribute images using a point-wise operator; the result is $\mathcal{A}_{merge}$.
Step 5. Compute the trees of shapes of $\mathcal{A}_{merge}$, that is, our tree of shapes of the color input image.

Before explaining the rationale behind this method, let us recall that any tree $\mathcal{T}$ valuated with a strictly increasing attribute a can be rebuilt by computing the min-tree of the attribute image $\mathcal{A}$. Indeed, any cut $CC([\mathcal{A} < \lambda])$ matches in the same time a single node in $\mathcal{T}$ (because a is increasing) and a node of in the min-tree of $\mathcal{A}$ (by lower cuts based definition of min-trees). As a result, instead of merging nodes in the original color space that would lead to overlapping components, merge takes place in an attribute space. The final tree of shapes is recovered from the tree of shapes of the attribute image. $Area$ attribute is a simple algebraic criterion—hence totally uncorrelated with the color space—that preserves shape inclusion semantics, e.g., a shape s_1 can only be included in s_2 if $\mathcal{A}(s_1) < \mathcal{A}(s_2)$. Let us now explain, how attribute images are merged together. For each point p, we need to select a scalar value from the triplet $(\mathcal{A}_r(p), \mathcal{A}_g(p), \mathcal{A}_b(p))$. While being purely algebraic, an order on the area values is meaningless, since a large area is not necessarily a good criterion to render shape significance. Since most problems arise when choosing the most significant channel on contours, we propose to merge the trees with gradient guidance. The red, blue, green gradients of each original images are denoted ∇_r, ∇_g and ∇_b. The resulting attribute image is composed from the pixels of the attributes images with the highest gradient magnitude:

$$\mathcal{A}_{\text{merge}}(p) = \begin{cases} \mathcal{A}_r(p) & \text{if } |\nabla_r(p)| \geq \max(|\nabla_g(p)|, |\nabla_b(p)|) \\ \mathcal{A}_g(p) & \text{if } |\nabla_g(p)| \geq \max(|\nabla_r(p)|, |\nabla_b(p)|) \\ \mathcal{A}_b(p) & \text{if } |\nabla_b(p)| \geq \max(|\nabla_r(p)|, |\nabla_g(p)|) \end{cases}$$

Figure 4 shows a simple case where geometry information are held by the chrominance. It shows two regions A and B that belongs respectively to red and green channels. If we computed the tree of shapes using achromatic information only, it would detect $A \cup B$ as a single shape, but neither A nor B. Using nodes from the trees of shapes computed independently, we get three possible area values λ_1, λ_2, λ_3 that relate to regions Ω (the full domain), A and B (we assume $\lambda_1 > \lambda_2 > \lambda_3$). To merge the attribute images, we rely on the gradient which is null everywhere except on the regions' boundaries. The image $\arg\max\{|\nabla_r|, |\nabla_g|\}$ shows in red (resp. green) the location where the maximal gradient comes from the red (resp. green) channel and in white the location where red and green gradient are equals. Thus, pixels of the fusion attribute image A_{merge} in white regions can be computed in two ways: using area values coming either from red or green channel. Computing the tree of shapes of those

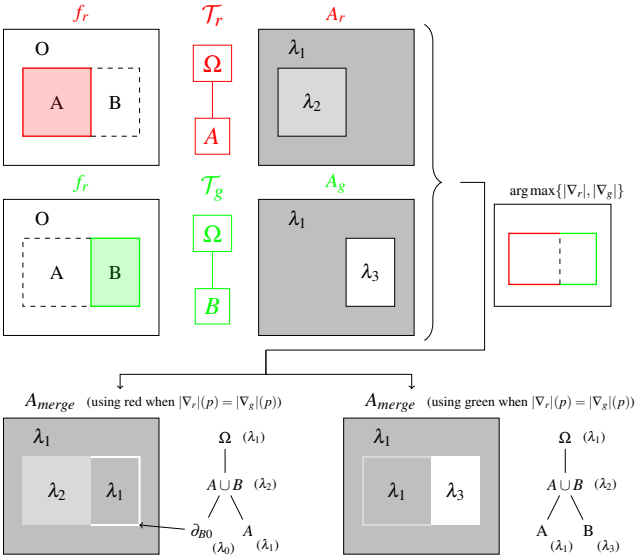


Fig. 4. Illustration of the proposed method on a multi-channel image where chrominance holds geometry.

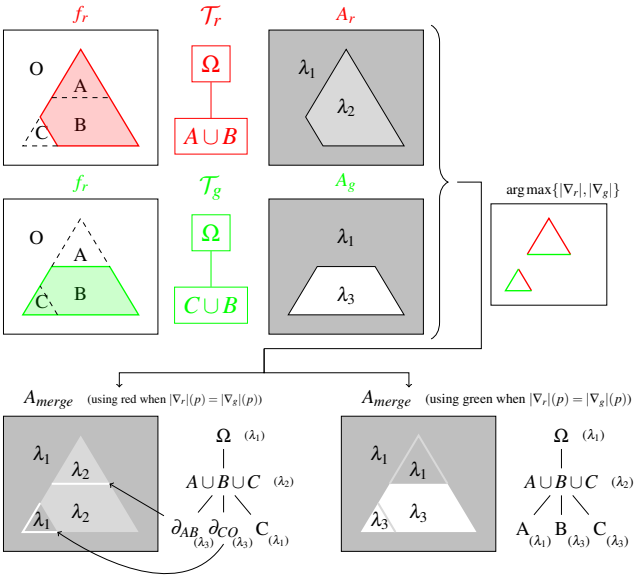


Fig. 5. Illustration of the proposed method on a multi-channel image where each channel taken independently holds incomplete information.

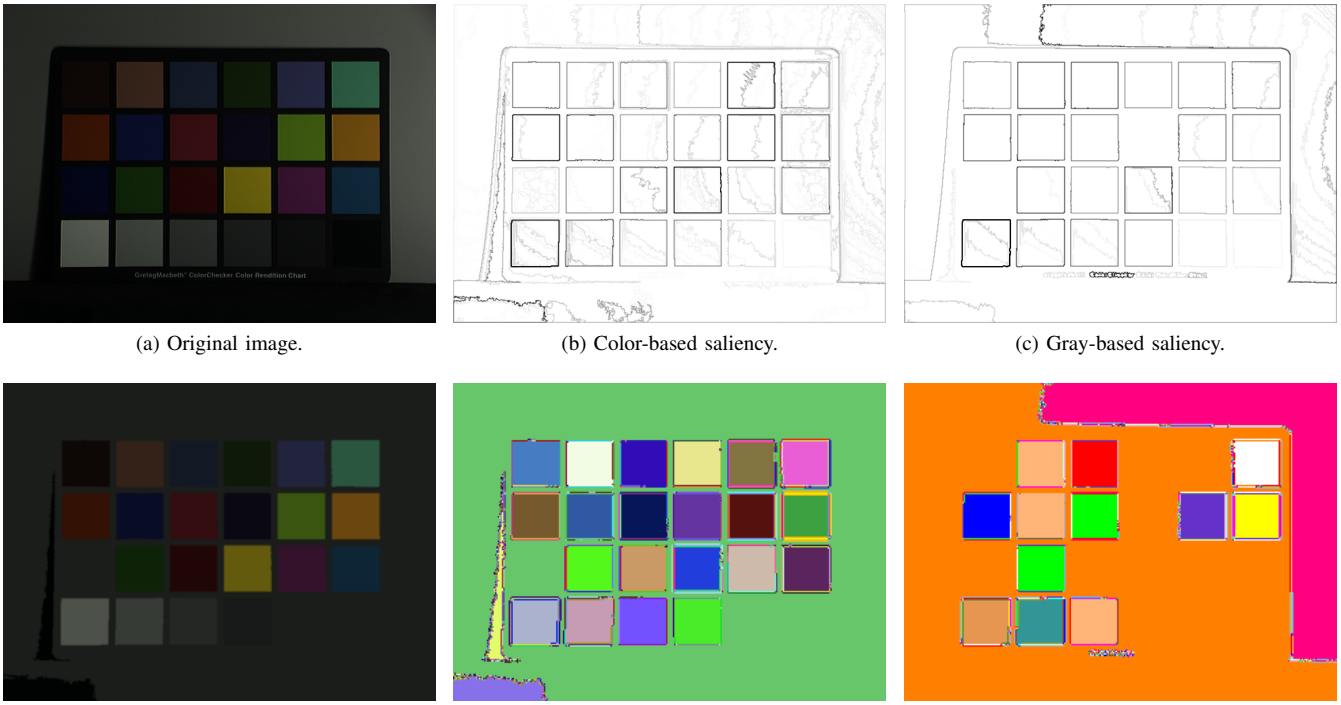
images afterward results in two different trees but both trees are able to distinct A and B regions.

Figure 5 shows a more complex case where geometry information is split across the channels of the image. The main object, a triangle, is composed by the regions A, B and C. The red channel of the image misses the C region, while the green channel misses the A region. Thus, the trees of shapes computed on those channels independently are not able to fully detect the object. When computing the area of each node, we get three possible values λ_1, λ_2 and λ_3 ($\lambda_1 > \lambda_2 > \lambda_3$) that relates to regions Ω (the full domain), $A \cup B$ and $B \cup C$ respectively. The same reasoning then holds as for the previous example. Afterward, computing the tree of shapes A_{merge} results in a tree holding a node that fully matches the triangle (the level line λ_2).

IV. Results

In the following, we compare an object detection method from Xu et al. [17] on both the tree of shapes of the luminance

Fig. 7. Results of simplification. First row: original images. Second row: simplification by thresholding the saliency map of the tree of shapes on luminance. Third row: simplification by thresholding (with the same value) the saliency map of the tree of shapes on colors.



(a) Original image. (b) Color-based saliency. (c) Gray-based saliency.

(d) Color-based simplification. (e) Color-based simplification (false colors). (f) Gray-based simplification (false colors).

Fig. 6. Saliency-based image simplification: the first two columns rely on the color tree of shapes; the rightmost column relies on the gray-level tree of shapes.

of the image and the tree of shapes of the color image computed with our proposed method. This method consists in computing a contextual energy composing three terms: the curvature that renders the regularity of a shape, a distance term between the internal and the external distribution around shape boundaries that measures how meaningful is a shape boundary and a constraint term that penalizes small components. The method parameters are the same for valuating nodes of the "gray level" and the color trees of shapes. Afterward, we select all shapes having local minimum energy value and we compute their extinction values. Since an extinction value reflects the persistence of the shape, we can then set this extinction value on shape boundaries to get a saliency map as shown in Figure 6. By thresholding this image, we can filter out some shapes such that only the most significant objects remain and leads to a simplification of the original image. In Figure 7, the remaining objects are filled with the mean of the component values.

Authors acknowledge that since most natural images have geometry information held by the luminance, images have been chosen such that luminance only does not allow to retrieve object shapes. Images are taken from the Berkley Segmentation Dataset [32] and from the Barnard's Color Constancy Dataset [33].

The calibration image in Figure 6 is a particularly good sample of images where chrominance matters, as it proves that using luminance only, some color squares cannot be distinguished from the background and merge rapidly with it. The corresponding saliency map does affirm this proposition. Also, our method tends to improve object detection in real case examples of natural images (see Figure 7) where still, the luminance misses geometry information.

V. Conclusion and Perspectives

In this paper we have presented a first proposal to compute a tree of shapes for color images whereas the equivalent notion of level lines, iso-color lines, cannot drive anymore the tree definition. The key idea behind our proposal is first to maintain some algebraic properties of shapes and second, to work in the attribute space instead of the original value space.

To obtain the tree of color shapes, we need to compute the trees of shapes on every image channels (Step 2, described in subsection III-B); for that we can rely on a quasi-linear algorithm given in [3]. Eventually the proposed method has a quasi-linear complexity, which makes it very usable in practice. It has been implemented within our C++ image processing library Olena [34] and it will be available in the next release.

The perspectives of the work presented here are related to the many applications relying on the tree of shapes. The preliminary experiments that we have performed show that the color tree of shapes is well-suited to applications such as filtering, simplifying, and segmenting color images. Some quantitative evaluations of the relevancy of the proposed tree will be the subject of a forthcoming extended paper.

References

[1] P. Salembier and M. Wilkinson, "Connected operators," *IEEE Signal Processing Magazine*, vol. 26, no. 6, pp. 136–157, 2009.

[2] L. Najman and H. Talbot, Eds., *Mathematical Morphology From Theory to Applications*. ISTE Ltd and John Wiley & Sons Inc, July 2010.

[3] T. Géraud, E. Carlinet, S. Crozet, and L. Najman, "A quasi-linear algorithm to compute the tree of shapes of

nd images," in *Proc. of Intl. Symp. on Mathematical Morphology*, ser. LNCS, vol. 7883, 2013, pp. 98–110.
[4] 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.
[5] V. Caselles and P. Monasse, "Grain filters," *JMIV*, vol. 17, no. 3, pp. 249–270, 2002.
[6] Y. Xu, T. Géraud, and L. Najman, "Morphological filtering in shape spaces: Applications using tree-based image representations," in *Proc. of ICPR*, 2012, pp. 485–488.
[7] C. Ballester, V. Caselles, L. Igual, and L. Garrido, "Level lines selection with variational models for segmentation and encoding," *JMIV*, vol. 27, pp. 5–27, 2007.
[8] A. Pardo, "Semantic image segmentation using morphological tools," in *IEEE Transactions on Image Processing*, 2002, pp. 745–748.
[9] J. Cardelino, G. Randall, M. Bertalmio, and V. Caselles, "Region based segmentation using the tree of shapes," in *Proc. of IEEE ICIP*, 2006, pp. 2421–2424.
[10] Y. Pan, "Top-down image segmentation using the mumford-shah functional and level set image representation," in *Proc. of IEEE ICASSP*, 2009, pp. 1241–1244.
[11] X.-S. Xia and J. Delon, "Shape-based invariant texture indexing," *IJCV*, vol. 88, no. 3, pp. 382–403, 2010.
[12] Y. Pan, J. Birdwell, and S. Djouadi, "Preferential image segmentation using trees of shapes," *IEEE Transactions on Image Processing*, vol. 18, no. 4, pp. 854–866, 2009.
[13] E. Meinhardt-Llopis, "Morphological and statistical techniques for the analysis of 3D images," Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, Spain, 2011.
[14] V. Caselles, B. Coll, and J. Morel, "Topographic maps and local contrast changes in natural images," *IJCV*, vol. 33, no. 1, pp. 5–27, 1999.
[15] F. Cao, P. Musé, and F. Sur, "Extracting meaningful curves from images," *JMIV*, vol. 22, pp. 159–181, 2005.
[16] 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.
[17] Y. Xu, T. Géraud, and L. Najman, "Context-based energy estimator: Application to object segmentation on the tree of shapes," in *Proc. of IEEE ICIP*, 2012, pp. 1577–1580.
[18] ——, "Salient level lines selection using the Mumford-Shah functional," in *Proc. of IEEE International Conference on Image Processing*, 2013.
[19] A. Victor, J. Roselin, and V. Kavitha, "Preferential image segmentation using j segmentation based on color, shape and texture," *International Journal of Engineering and Technology*, vol. 2, no. 2, pp. 131–135, 2010.
[20] B. R. Kiran, J. Serra *et al.*, "Global-local optimizations by hierarchical cuts and climbing energies," *Pattern Recognition*, vol. 47, no. 1, pp. 12–24, Jan. 2014.
[21] L. Najman and M. Schmitt, "Geodesic saliency of watershed contours and hierarchical segmentation," *IEEE Trans. on PAMI*, vol. 18, no. 12, pp. 1163–1173, 1996.
[22] P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval," *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 561–576, 2000.
[23] P. Soille, "Constrained connectivity for hierarchical image partitioning and simplification," *IEEE Trans. on PAMI*, vol. 30, no. 7, pp. 1132–1145, 2008.
[24] P.-E. Forssén, "Maximally stable colour regions for recognition and matching." IEEE, 2007, pp. 1–8.
[25] V. Barnett, "The ordering of multivariate data," *J. of the Royal Statistical Society*, vol. 139, pp. 318–355, 1976.
[26] J. Angulo, "Unified morphological color processing framework in a lum/sat/hue representation," in *Proc. of Intl. Symp. on Math. Morphology*, 2005, pp. 387–396.
[27] J. Angulo and J. Serra, "Modelling and segmentation of colour images in polar representations," *Image and Vision Computing*, vol. 25, no. 4, pp. 475–495, 2007.
[28] J. Angulo, "Morphological colour operators in totally ordered lattices based on distances: Application to image filtering, enhancement and analysis," *CVIU*, vol. 107, no. 1, pp. 56–73, 2007.
[29] O. Lezoray, C. Charrier, A. Elmoataz *et al.*, "Rank transformation and manifold learning for multivariate mathematical morphology," in *Proc. of European Signal Processing Conference*, vol. 1, 2009, pp. 35–39.
[30] S. Velasco-Forero and J. Angulo, "Supervised ordering in $\mathcal{R}_p$: application to morphological processing of hyperspectral images." *IEEE Transactions on Image Processing*, vol. 20, no. 11, p. 3301, 2011.
[31] N. Passat and B. Naegel, "An extension of component-trees to partial orders," in *Proc. of IEEE International Conference on Image Processing*, 2009, pp. 3981–3984.
[32] 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 *Proc. of International Conference on Computer Vision*, vol. 2, July 2001, pp. 416–423.
[33] K. Barnard, L. Martin, B. Funt, and A. Coath, "A data set for color research," *Color Research & Application*, vol. 27, no. 3, pp. 147–151, 2002.
[34] R. Levillain, T. Géraud, and L. Najman, "Why and how to design a generic and efficient image processing framework: The case of the Milena library," in *Proc. of IEEE International Conference on Image Processing*, 2010, pp. 1941–1944, http://olena.lrde.epita.fr .