MToS: A Tree of Shapes for Multivariate Images

Edwin Carlinet*, *Student Member, IEEE*, Thierry Géraud, *Member, IEEE*

***Abstract*—The topographic map of a gray-level image, also called tree of shapes, provides a high-level hierarchical representation of the image contents. This representation, invariant to contrast changes and to contrast inversion, has been proved very useful to achieve many image processing and pattern recognition tasks. Its definition relies on the total ordering of pixel values, so this representation does not exist for color images, or more generally, multivariate images. Common workarounds such as marginal processing, or imposing a total order on data are not satisfactory and yield many problems. This paper presents a method to build a tree-based representation of multivariate images which features marginally the same properties of the gray-level tree of shapes. Briefly put, we do not impose an arbitrary ordering on values, but we only rely on the inclusion relationship between shapes in the image definition domain. The interest of having a contrast invariant and self-dual representation of multivariate image is illustrated through several applications (filtering, segmentation, object recognition) on different types of data: color natural images, document images, satellite hyperspectral imaging, multimodal medical imaging, and videos.**

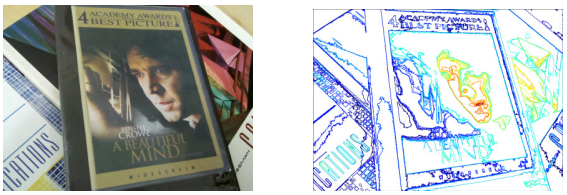
***Keywords*—Tree of shapes, connected operators, mathematical morphology, multivariate images.**

I. Introduction

THE topographic map, also called Tree of Shapes (ToS), is a hierarchical representation of a gray-level image in terms of the inclusion of its level lines [1, 2]. Every node of the ToS represents a connected component whose border is a level line and, typically, the number of nodes is close to the number of pixels for a non-degenerated image. The ToS of an image thus offers a description of the image contents as a collection of connected components, structured as a tree thanks to the inclusion of these components. Surprisingly this rich structure can be computed efficiently [3, 4], and can be also efficiently stored in memory [5]. Representing an image by a tree of its components means relying on a versatile and simple structure, and the abstraction offered by this structure enables to perform advanced image processing tasks in a very simple way [5, 6]. For example, simplifying an image can be performed by removing some nodes, identifying an object by selecting the most suitable node, and so on. Over the past decade, this representation has been used in a wide range of applications: denoising [7], filtering [8, 9], local feature detection [10], texture indexing [11], classification [12], segmentation [7, 13], tracking [14], and scene matching [15].

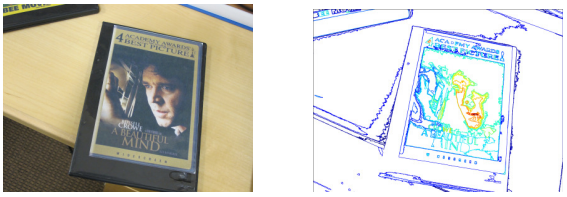
If the ToS achieves good results in applications, it is due to the mathematical properties held by this representation [16].

* edwin.carlinet@lrde.epita.fr
The authors are both with EPITA Research and Development Laboratory (LRDE), 14-16 rue Voltaire, FR-94276 Le Kremlin-Bicêtre, France, and with Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, A3SI, ESIEE Paris, 2 bd Blaise Pascal, B.P. 99, FR-93162 Noisy-le-Grand Cedex, France.



(a) An image containing a film cover (left) and some meaningful level lines extracted from its MToS (right).

(b) The pixel values of the original image (a) have been modified by a global marginal inversion and a change of contrast (left), and by a local change of contrast (right). These two new images have the same MToS and thus the same level lines as the original image.



(c) A different view of the film cover contained in (a). The level lines of this new image globally match the ones of (a), which means that the MToS properly encodes the contents of both images.

Fig. 1. Robustness of the level lines w.r.t. transformations in the value space (b) and the domain space (c). The depicted level lines are selected using the Maximum Stability criterion (as for the MSER) and colorized w.r.t. their level of inclusion in the MToS.

1. It is based on the inclusion of connected components, so filtering this tree by removing nodes leads to connected operators [17], *i.e.* operators that do not shift the object boundaries. **2.** It is invariant by any contrast change [1]. Schematically, only the ordering of gray-level values matters (not the absolute values); as a consequence, it allows to handle poorly contrasted objects. It is also robust to *local* changes of contrast, as we expect the level lines to remain globally the same (see Fig. 1b). **3.** It is a self-dual representation of the image. This feature is fundamental in a context where structures may appear both on a brighter background or on a lighter one, or if we cannot (or do not want) make any assumption about the object / background layout. Self-dual operators have the ability to deal with both dark and light

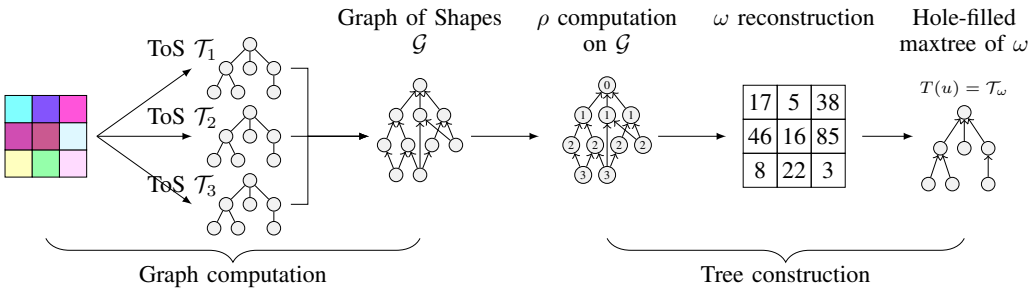


Fig. 2. The 5-steps process of the proposed method. (1) The input image u is decomposed into individual channel $u_1, u_2, \dots u_n$, for which the ToS's are computed, (2) the ToS's are merged into the Graph of Shapes (GoS) $\mathcal{G}$ (2), an algebraic attribute is computed on $\mathcal{G}$ and, (4) yields a scalar attribute map ω, (5) a final tree is built upon ω.

objects in a *symmetric* way [18, 19]. **4.** It allows a simple multi-scale analysis of the image since the shapes are organized in a tree w.r.t. their inclusion. **5.** The level lines describe the object boundaries in a non-local way. Contrary to many key-point detectors which rely on local information, level lines may be large closed curves, tangent to the gradient, and fitting object contours [20].

While the ToS is well defined on gray-scale images, it is not the case with multivariate data. Indeed, like most morphological trees (*e.g.*, min and max-trees), the ToS relies on an ordering relation on values which has to be total. If it is not, the shapes yield components that overlap, and we do not have an inclusion tree. Handling correctly multivariate data is of main interest in many fields of image processing. The most prominent example is a sensitive color processing of natural images, but many other types of images are multivariate: satellites provide multi-spectral or hyper-spectral images with hundreds of bands, medical processing provide multi-modal images acquired by several devices, and so on. Therefore, to overcome this problem, most authors have been focusing on defining a total order on multivariate data. However, from our point of view, the most important concept in the morphological trees lies in the inclusion of shapes. So, following this idea, this paper details a novel approach which does not intent to build a total order, but tries instead to build up a set of non-overlapping shapes from an arbitrary set of shapes using the inclusion relation only.

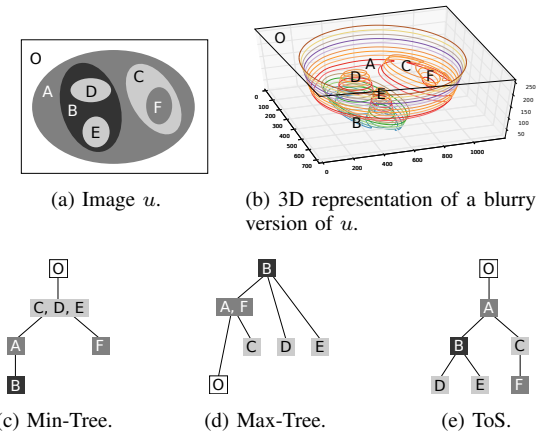
The method to build a ToS for multivariate images is a simple 5-step process, which consists basically in two parts depicted in Fig. 2. The first part is the construction of a Graph of Shapes (GoS) from ToS's computed marginally on each component (steps 1–2). The second part aims at deducing a tree from the GoS and consists in computing a tree over an image reconstructed from an attribute valuated on the GoS (steps 3–5). The result of the process is a morphological tree which features the invariance to any *marginal* change or inversion of contrast. To the best of our knowledge, this is the first time a multivariate image representation featuring such property has been proposed (please note that we have previously sketched this representation in a short paper [21]).

The paper is organized as follows. In Section II, we remind the basic principles about the ToS, and we explain in Section III the main lines of the method extending this tree to handle multivariate data. Section IV exposes and proves the properties held by the MToS. Section V compares our work with some other classical approaches. Section VI shows some applications highlighting the versatility and the potential of the MToS. Last Section VII concludes and gives some perspectives of our work.

As an effort to support reproducible research, some supplementary materials for this present paper (images, binaries, demos, and source code) are available online [22].

II. Definition of the Tree of Shapes (ToS)

A. Mathematical Background and Definitions



(a) Image u. (b) 3D representation of a blurry version of u.

(c) Min-Tree. (d) Max-Tree. (e) ToS.

Fig. 3. An image (a), and its morphological component trees (c) to (e).

Let an image $u : \Omega \to E$ defined on a domain Ω with values on a set E embedded with an ordering relation $\leq$. Let, $[u < \lambda]$ (resp. $[u > \lambda]$) with $\lambda \in \mathbb{R}$ be a threshold set of u (also called respectively lower and upper level sets) defined as:

$$[u < \lambda] = \{x \in \Omega,\, u(x) < \lambda\}. \tag{1}$$

We note $\mathcal{CC}(X), X \in \mathcal{P}(E)$ the set of connected components of X using the classical arcwise connectivity (the 4- or 8-connectivity for 2D images). 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, defined on upper level sets, is called the *max-tree*. These trees are illustrated in Figs. 3c and 3d, where each node is denoted by its own ("proper") pixels, *i.e.*, the pixels that are not present in any of its descendants. For example, in Fig. 3d(c), the node "A,F" actually represents the component $A \cup F \cup O \cup C$.

Let us consider the hole-filling operator defined by $\mathcal{H}(X) = \Omega \setminus \mathcal{CC}_{\partial\Omega}(\Omega \setminus X)$, where $\mathcal{CC}_{\partial\Omega}(Y)$ denotes the connected components of Y that connects with the image border. We call a *shape* any element of

$$\mathcal{S} = \{\mathcal{H}(\Gamma),\, \Gamma \in \mathcal{CC}([u < \lambda])\}_\lambda \cup \{\mathcal{H}(\Gamma),\, \Gamma \in \mathcal{CC}([u > \lambda])\}_\lambda. \tag{2}$$

Given $\mathcal{S}$ and a shape $A \in \mathcal{S}$, we note $A\uparrow = \{X \in \mathcal{S},\, A \subseteq X\}$, which is the set of ancestors of A.

If $\leq$ is total, any two shapes are either disjoint or nested, hence the cover of $(\mathcal{S}, \subseteq)$ forms a tree called the *Tree of Shapes (ToS)* (see Fig. 3e). In the rest of the paper, we implicitly consider the cover of $(\mathcal{S}, \subseteq)$ while writing $(\mathcal{S}, \subseteq)$ only. The level lines of u are the contours of the shapes. Using the image representation given in [3] and briefly reviewed in Section II-B, one can ensure each level line is an isolevel closed curve given that $\leq$ is a total order. Note that the ToS encodes the shapes inclusion but also the level lines inclusion that are the contours of the shapes. Without loss of generality, we will consider $E = \mathbb{R}^n$ throughout this paper, and we will note u for scalar images ($n = 1$) and $\mathbf{u}$ for multivariate ones ($n > 1$).

A transformation ψ is said contrast change invariant if given a strictly increasing function $g : \mathbb{R} \to \mathbb{R}$, $g(\psi(u)) = \psi(g(u))$. Moreover, the transformation is said self-dual if it is invariant w.r.t. the complementation *i.e.* $\complement(\psi(u)) = \psi(\complement(u))$ (for images with scalar values $\complement(u) = -u$). When ψ is both self-dual and contrast change invariant, then for any strictly monotonic function G (*i.e.*, either strictly increasing or decreasing), we have $G(\psi(u)) = \psi(G(u))$.

The ToS is actually a support for many self-dual morphological operators so this tree representation of images is said to be self-dual, and it verifies $T(-u) = T(u)$. It is also a morphological representation since it verifies $T(G(u)) = T(u)$ for any strictly monotonic function G.

B. Topological Considerations

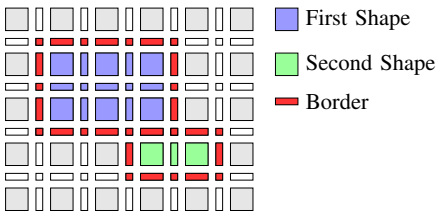


Fig. 4. Shapes on the cubical grid (here the 2D square grid).

Throughout this paper, we assume that the image has its domain on a cubical grid that allows continuous properties while staying on a discrete space. The algorithm proposed by Géraud et al. [3] to compute the ToS in grayscale uses this representation as well and more advanced details about topological properties with this grid can be found in [23]. We simply recall basic notions that will be necessary for the proofs of the present paper. We note $\mathcal{K}_\Omega$ the domain Ω immersed on the cubical 2D grid. In the previous sections, as a matter of clarity, we have noted Ω the domain of the image, but $\mathcal{K}_\Omega$ was always assumed. On the Fig. 4, original pixels are represented by 2-faces (large square) and intermediate pixels are added (1-faces and 0-faces). In the ToS, a *shape* A is an open set on the grid and may be composed of 0-, 1-, and 2-faces. In Fig. 4, blue and green elements represent two disjoint shapes A and B. On the other hand, the border of a shape is composed of 0- and 1-faces only (red elements). Shapes boundaries are the actual level lines of the image. We note ∂A, the border of the set A and $\bar{A} = A \cup \partial A$ the closure of A. Note that in the tree of shapes, two shapes are either nested or disjoint but $\bar{A}$ and $\bar{B}$ may overlap as shown in Fig. 4.

C. The ToS Computation Algorithm

In [3], the authors propose a quasi-linear algorithm to compute the ToS of nD images. This algorithm has two main steps. The first one is a sorting step, that aims at ordering the pixels from the root to the leaves. For that, it relies on a propagation front starting from the border, and browsing the image contents "continuously". The continuity in the space domain is naturally handled by the spatial consistency of the front. The continuity in the value domain is ensured by the combination of a hierarchical queue associated with the front and by the 1-faces, which allow to represent that different level lines can pass between adjacent pixels. On the simple image of Fig. 3a, a sorting result can be the sequence of pixels of the regions O, A, C, B, E, F, and D. The second main step of the algorithm is a Tarjan's union-find that processes the pixels in the reverse order, and builds the tree from the leaves up to the root. This step can been roughly seen as an immersion process from the "deepest" extrema of the image.

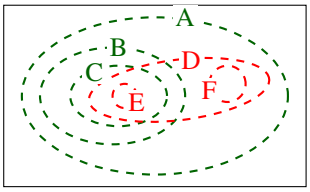
The way this algorithm acts has actually inspired our approach to extend the ToS to multivariate images. Indeed, in the next section, we will aim at defining an ordering on the pixels that renders their level of inclusion.

III. The Multivariate Tree of Shapes (MToS)

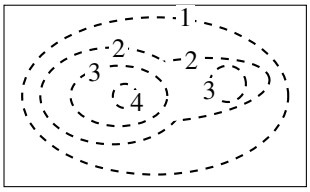
A. Method Description

Let us first relax the definition of *shape*. A *shape* X is a connected component of Ω without holes (*i.e.*, such that $\mathcal{H}(X) = X$).

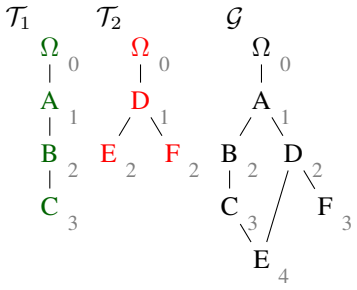
Given a family of shape sets, namely $\mathcal{M} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n\}$, where each element $(\mathcal{S}_i, \subseteq)$ forms a tree, we note $\mathcal{S} = \bigcup \mathcal{S}_i$ the initial shape set. Note that $(\mathcal{S}, \subseteq)$ generally does not form a tree but a graph since shapes may overlap. We aim at defining a new set of shapes $\boldsymbol{\mathcal{S}}$ such that any two shapes are either nested or disjoint. We do not constrain $\boldsymbol{\mathcal{S}} \subseteq \mathcal{S}$, *i.e.*, we allow the method to build new shapes that were not in the original shape set, *e.g.*, by



(a) Original image u (2-channels) and its shapes.



(c) ω image built from $\mathcal{G}$.



(b) The marginal ToS's $\mathcal{T}_1$, $\mathcal{T}_2$ and the GoS. The depth appears in gray near the nodes.

(d) The max-tree $\mathcal{T}_f$ of ω.

Fig. 5. The method illustrated on a very simple example.

merging some of them. We note $T : \mathbb{R}^{n\Omega} \to (\mathcal{P}(\mathcal{P}(\Omega)), \subseteq)$ the process that builds a tree of shapes $(\boldsymbol{S}(\mathbf{u}), \subseteq)$ from an image $\mathbf{u} \in \mathbb{R}^{n\Omega}$.

We propose a method in 5-steps (see Fig. 2) which has two main objectives: merging many trees into a single graph structure, and the deduction of a tree from this graph.

1) The Graph of Shapes (GoS): First $\mathbf{u}$ is decomposed in individual channels $u_1, u_2, \ldots, u_n$ for which the ToS $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n$ are associated with the shape sets $\mathcal{S}_1, \mathcal{S}_2, \ldots \mathcal{S}_n$. Let $\mathcal{S} = \bigcup \mathcal{S}_i$, we call the GoS $\mathcal{G}$ the cover of $(\mathcal{S}, \subseteq)$, *i.e.*, it is the inclusion graph of all the shapes computed marginally.

With "standard" morphological hierachies (min-/max- trees) and their extension (the component-graph [24, 25]), for any point x, there exists a *single* smallest component that contains x. As a consequence, a point belongs to a *single* node in the structure. In the GoS, a point may belong to several nodes which are not in the same lineage. For example, in Fig. 5, the points in $(B \cap D) \setminus C$ belong to both nodes B and D. It leads to a weird paradox, even if the GoS is actually a tree, it is not a valid morphological tree as two shapes overlap without being nested. Thus, we cannot just extract a tree (*e.g.* with the minimum spanning tree) from the GoS as it will not be valid.

2) Deducing a Tree from the GoS: Let $\rho : \mathcal{P}(\Omega) \to \mathbb{N}$ be an algebraic decreasing shape attribute, *i.e.*, $\forall A, B \in \mathcal{S},\ A \subset B \ \Rightarrow\ \rho(A) > \rho(B)$.

We will discuss in the next section the choice of ρ and why we consider the *depth* attribute. The *depth* of a shape in $\mathcal{G}$ is the length of the longest path of a shape A from the root. Let $\omega : \Omega \to \mathbb{N}$ defined as:

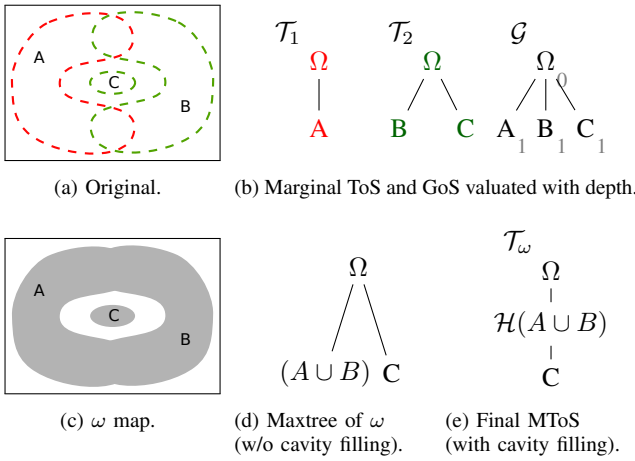
$$\omega(x) = \max_{X \in \mathcal{S}, x \in X} \rho(X). \tag{3}$$

The map ω associates for each point x, the depth of the deepest shape containing x (see Figs. 5b and 5c). Let $\mathbb{C} = \{\mathcal{CC}([\omega \geq h]), h \in \mathbb{N}\}$. $(\mathbb{C}, \subseteq)$ is actually the max-tree of ω. Finally, we consider $\boldsymbol{S} = \mathcal{H}(\mathbb{C})$ and $(\boldsymbol{S}, \subseteq)$ as the final MToS $\mathcal{T}_\omega$ (see Fig. 5d). Because $\mathbb{C}$ may form component with holes, the hole-filling ensures that components are valid shapes (see Fig. 6). Note that we illustrate in Fig. 5 the fact that new shapes may be created that where not in the original shape set as $B \cup D$ belong neither to $\mathcal{T}_1$ nor to $\mathcal{T}_2$.

3) Rationale: Let us now explain the rational behind the method. First, we start with computing the marginal ToS's of u that give us an initial shape set. The multiple trees provide a representation of the original image and u can be reconstructed marginally from them. However, handling several trees is not straightforward and they lack an important information: how the shapes of one tree are related (w.r.t. the inclusion) to the shapes of the other trees. The graph $\mathcal{G}$ is nothing more than these trees merged in a unique structure that adds the inclusion relation that was missing previously. As consequence, $\mathcal{G}$ is "richer" than $\{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$ and because the transformation from $\{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$ to $\mathcal{G}$ is reversible, $\mathcal{G}$ is a complete representation of $\mathbf{u}$ (*i.e.* $\mathbf{u}$ can be reconstructed from $\mathcal{G}$). Moreover, $\mathcal{G}$ is also a self-dual and a contrast invariant representation of $\mathbf{u}$ because $\{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$ are.

The second part of the method tries to extract a tree from $\mathcal{G}$ verifying the constraints given in Section IV-A. The key issue is to get a new set of shapes from $\mathcal{G}$ that do not overlap. The first observation is that for any decreasing attribute ρ, then $(\mathcal{S}, \subset)$ is isomorphic to $(S, \mathcal{R})$ where $A \mathcal{R} B \Leftrightarrow \rho(A) > \rho(B)$ and $A \cap B \neq \emptyset$. This just means that the inclusion relationship between shapes that we want to preserve can be expressed in terms of a simple ordering relation on $\mathbb{R}$ with the values of a decreasing attribute. Suppose now that $(\mathcal{S}, \subset)$ is a tree and consider the image $\omega(x) = \max_{x \in X, X \in \mathcal{S}} \rho(x)$, we thus have $\mathbb{C} = \{\mathcal{CC}([\omega \geq h]), h \in \mathbb{N}\} = \mathcal{S}$. In other words, the max-tree of ω reconstructed from ρ valuated on a tree $\mathcal{T}$ yields the same tree and more generally, if a shape A do not overlap any other shape, it belongs to $\mathcal{CC}([\omega \geq h])$. In the next section, we explain the reason to choose the *depth* for ρ.

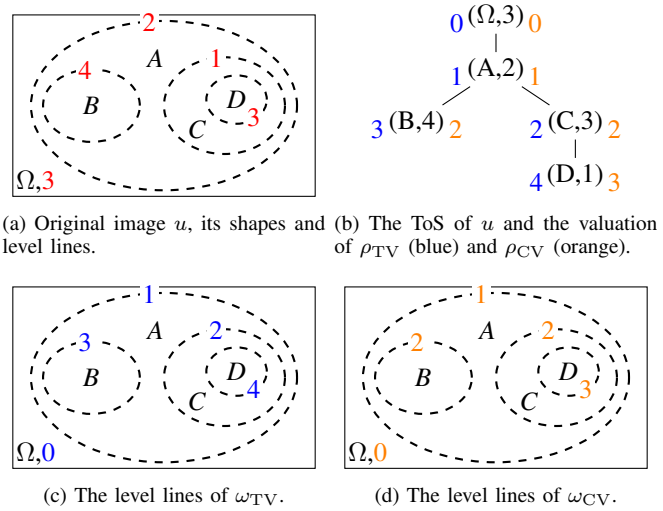
4) Image reconstruction from the MToS: Whereas $\mathcal{G}$ is a complete representation of $\mathbf{u}$, the ToS is not, so $\mathbf{u}$ cannot be reconstructed from it. Indeed, the tree construction process merges some marginal shapes (in the most sensible way as possible). Consequently a node of the final tree gets associated with multiple values of $\mathbf{u}$. Actually this problem is not new; it exists when extending the min- and max- trees with a total preorder, where the loss of the anti-symmetry implies that some "equivalent" values belong to the same node. Naegel and Passat [26], and later Tushabe and Wilkinson [27], have introduced some reconstruction strategies to solve the assignment problem. The main idea is to associate to a node a single value computed from the set of values it contains. For example, in [26] the authors proposed to assign the average vector or the median vector to the node. In [27], the same principle is applied but only to the pixels that belong to the nodes that are filtered out; the values of the other pixels remain unchanged. The authors also proposed two others strategies, assigning the



(a) Original. (b) Marginal ToS and GoS valuated with depth.

(c) ω map. (d) Maxtree of ω (w/o cavity filling). (e) Final MToS (with cavity filling).

Fig. 6. On the need for the saturation.



(a) Original image u, its shapes and level lines. (b) The ToS of u and the valuation of ρ_{TV} (blue) and ρ_{CV} (orange).

(c) The level lines of ω_{TV}. (d) The level lines of ω_{CV}.

Fig. 7. Equivalence between the level lines of a gray-level image u and the level lines of the distance maps ω_{TV} and ω_{CV}.

closest pixel's value from the last surviving parent, where "closest" can be interpreted in the value space (1st strategy) or in the domain space (2nd strategy). In the present paper, precisely in Section VI, to reconstruct an image from a MToS, we use the strategy to assign to each node the average vector value from the original image.

B. Choosing a Sensible ρ Function

The 3rd step of the method involves choosing an attribute to be computed over the GoS $\mathcal{G}$. This is a critical step since it decides which shapes are going to be merged or removed.

1) Level-Lines as a Distance Problem: Consider the distance between two points (p, p') in Ω:

$$d_{\mathrm{TV}}(p,p') = \min_{C_{(p,p')}} \int_0^1 |\nabla u(C(t)).\dot{C}(t)|\, dt, \qquad (4)$$

where $C(t)$ is a path in Ω from p to p'. Equation (4) is actually the minimum total variation (TV) along a path between p and p'. This measure has been used by Dubrovina et al. [28] for segmenting where the ToS is used as a front-end to compute efficiently the level set distance. Let $\omega_{\mathrm{TV}}(x) = d_{\mathrm{TV}}(\partial\Omega, x)$ be the Total Variation distance map from the border. This distance map can be computed using a simple decreasing attribute on the ToS by summing the variations from the root to the nodes. Then, instead of considering the tree $\mathcal{T}$ of u level lines, one can consider the max-tree $\mathcal{T}_\omega$ of equidistant lines. Both are equivalent in gray-level (by prop. 2).

The problem with the Total Variation metric lies in that it depends on u, *i.e.*, ω_{TV} is not contrast invariant. A contrast invariant counterpart would be to only count the number of variation, *i.e.*, the minimum number of level lines to traverse to get p:

$$d_{\mathrm{CV}}(p,p') = \min_{C_{(p,p')}} \int_0^1 \mathbb{1}\{\nabla u(C(t)).\dot{C}(t)\}\, dt. \qquad (5)$$

Algorithmically speaking, building ω_{CV} consists in computing the depth attribute $\rho_{\mathrm{CV}}(A) = |A\uparrow|$ and reconstructing $\omega_{\mathrm{CV}}(x) = \max_{X \in \mathcal{S}, x \in X} \rho_{\mathrm{CV}}(X)$. This process is shown on Fig. 7.

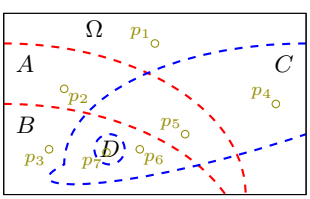
2) Distance Map with Multivariate Images: Based on the equivalence between level lines and equidistant lines in gray levels, one can produce a distance map for multivariate images. As with Eq. (5) the idea is to count the number of marginal level lines to traverse. Depending on the way we count the level lines, the distance map may have several semantics:

D_1: $\omega_{\mathrm{CV}}(x) = |X \in \mathcal{S}, x \in X| - 1$
counts the minimal number of marginal level lines we need to traverse to get x from the border.

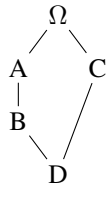
D_2: $\rho(A) = |A\uparrow|$ and $\omega_{\mathrm{CV}}(x) = \max_{X \in \mathcal{S},\, x \in X} \rho(X)$
counts the number of marginal level lines to traverse to get the deepest shape that contains x.

D_3: $\rho(A) = \max_{\phi \in [\Omega \rightsquigarrow A)} |\phi|$ and $\omega_{\mathrm{CV}}(x) = \max_{X \in \mathcal{S},\, x \in X} \rho(X)$
where $[\Omega \rightsquigarrow A)$ stands for the set of paths from the root to A in $\mathcal{G}$; it counts the number of marginal level lines *that are nested* to traverse to get the deepest shape that contains x.

These measures can be computed efficiently from $\mathcal{G}$ using basic graph algorithms (*e.g.*, shortest path algorithm for D_3 and ancestors counting algorithms for D_1 and D_2). The differences between them are shown in Fig. 8. While, the D_2 and D_3 distances yield a union of shapes only, the D_1 distances enables to get both union and intersection of shapes. However, in practice, they define similar shape sets that differ essentially for very small components (*i.e.*, at the noise level). This is illustrated in Fig. 9, which shows the ω map computed with the three distances, using a "heat" look-up table. As one can see, there are strong similarities between the three maps. Note that (e) seems to have less details than (c) and (d); this is due to the fact that the measure increases slower with D_3.

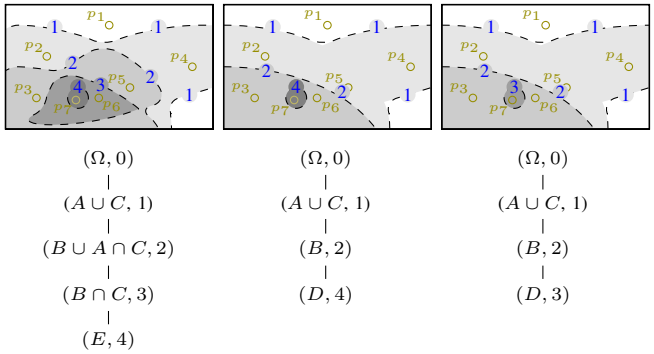


(a) Original image u, its shapes and level lines.



	D_1	D_2	D_3
p_1	0	0	0
p_2	1	1	1
p_3	2	2	2
p_4	1	1	1
p_5	2	1	1
p_6	3	2	2
p_7	4	4	3

(b) Graph of shapes $\mathcal{G}$ and measures D_1 D_2 and D_3 valuated on markers.



(c) The distance maps ω computed with D_1, D_2 and D_3 (from left to right) and their corresponding max-trees below.

Fig. 8. Differences between D_1, D_2, D_3 for the distance map computation.

However the absolute distance value between two shapes does not really matter: as soon as such a difference exists, the shapes appear in the final tree. As a consequence, for the illustrations in Section VI, we consider the distance D_3 as it is the fastest to compute. Note that it might be tempting to compute the distance map ω using a more conventional shortest-path algorithm, however it has been shown by Dubrovina et al. [28] that it cannot be used reliably to compute level line distance due to topological issues. It would yield a totally different set of level lines that do not represent correctly the image content.

IV. Properties of MToS

A. Properties

The method T produces $T(\mathbf{u}) = (\boldsymbol{S}(\mathbf{u}), \subseteq)$ having the following properties:

(P0) *Well-formed tree* On "classical" images (not synthetic), the tree has a sufficient height and a sufficient number of nodes. In other words, it produces a tree which is topologically similar to the "classical" ToS.

(P1) *Domain covering* $\left(\bigcup_{X \in \boldsymbol{S}(\mathbf{u})} X\right) = \Omega$
(a point belongs to one shape at least)

(P2) *Tree structure*
$\forall X, Y \in \boldsymbol{S}(\mathbf{u})$, either $X \cap Y = \emptyset$ or $X \subseteq Y$ or $Y \subseteq X$
(any two shapes are either nested or disjoint)

(P3) If a shape $X \in \mathcal{S}$ verifies:
$$\forall Y \neq X \in \mathcal{S},\ X \cap Y = \emptyset \text{ or } X \subset Y \text{ or } Y \subset X$$
then $X \in \boldsymbol{S}(\mathbf{u})$ (any shape that does not overlap with any other shape exists in the final shape set). A corollary of this property is the *scalar ToS equivalence*. If $\mathcal{M} = \{\mathcal{S}_1\}$ then $\boldsymbol{S}(\mathbf{u}) = \mathcal{S}_1$ *i.e.* for scalar images, the tree built by the method is equivalent to the gray-level ToS.

(P4) *Marginal contrast change/inversion invariance.*
Let us consider $\mathbf{G}(\mathbf{u}) = (G_1(u_1), G_2(u_2), \ldots, G_n(u_n))$, where G_i is a strictly monotonic function, then T is invariant by marginal inversion/change of contrast, that is, $T(\mathbf{G}(\mathbf{u})) = T(\mathbf{u})$.

B. Proof of Correctness

We now prove that the MToS construction process verifies the properties given in Section IV-A.

Proposition 1. *Let a shape $A \in \mathcal{S}$, and $x \in \partial A$, then $\forall X \in \mathcal{S},\ x \in X \ \Rightarrow\ X \in A\uparrow$.*

Proof: Suppose $x \in \partial A$, and a shape $B \in \mathcal{S}$ such that $x \in B$. Then, B is an open set, so it contains the 2-face in A adjacent to x and $B \cap A \neq \emptyset$. Two shapes being either disjoint or nested, we have $B \subseteq A$ or $A \subseteq B$. Since $x \in B$ but $x \notin A$, it follows that $A \subseteq B$ and $B \in A\uparrow$. ■

Proposition 2. *Given a set of shapes $\mathcal{S}$ where any two shapes are either disjoint or nested ($(\mathcal{S}, \subseteq)$ is a tree) and a strictly decreasing attribute ρ then $\mathbb{C} = \mathcal{S}$*

Proof: ($\Rightarrow$) Let a shape $A \in \mathcal{S}$. $\forall x \in A$, we have:
$$\omega(x) = (\max_{X \in \mathcal{S}, x \in X} \rho(X)) \geq \rho(A)$$
Let now, $x \in \partial A$, $\forall X \in \mathcal{S}, x \in X$, we have $A \subset X$ (by prop. 1), thus $\rho(X) < \rho(A)$ and $\omega(x) < \rho(A)$. It follows that $A \in \mathcal{CC}([\omega \geq \rho(A)])$ and $A \in \mathbb{C}$.
($\Leftarrow$) Let $A \in \mathcal{P}(\Omega), A \notin \mathcal{S}$, we note $\mathrm{SES}(A)$ the smallest enclosing shape that includes A. Suppose now that $A \in \mathbb{C}$, then $\exists \lambda \in \mathbb{N}, A \in \mathcal{CC}([\omega \geq \lambda])$. Let $\alpha \in \mathbb{N}$, $\alpha = \min_{x \in A} \omega(x)$, then $\lambda \leq \alpha$. Yet,

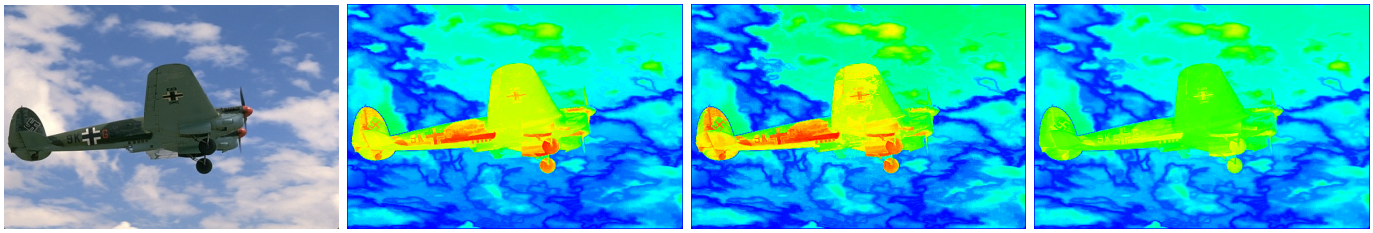
$$\begin{aligned}\alpha = \min_{x \in A} \omega(x) &= \min_{x \in A} \max_{X \in \mathcal{S},\, x \in X} \rho(X) \\ &= \min_{X \in \mathcal{S},\, A \cap X \neq \emptyset} \rho(X) \\ &= \rho(\mathrm{SES}(A))\end{aligned}$$

Thus, $A \in \mathcal{CC}([\omega \geq \lambda])$ with $\lambda \leq \rho(\mathrm{SES}(A))$. But, since $A \subsetneq \mathrm{SES}(A)$ and $A, \mathrm{SES}(A) \in \mathbb{C}$ then $\lambda > \rho(\mathrm{SES}(A))$ which contredicts $\lambda \leq \rho(\mathrm{SES}(A))$. Therefore, $A \notin \mathcal{S} \Rightarrow A \notin \mathbb{C}$ ■

A direct consequence of prop. 2 is that given a ToS $\mathcal{T}$ and the max-tree $\mathcal{T}_\omega$ of the image ω reconstructed from a decreasing attribute ρ over $\mathcal{T}$, then $\mathcal{T}_\omega = \mathcal{T}$.

Proposition 3. *The method provides a tree $\mathcal{T}_\omega$ that verifies the property (P1), (P2), (P3), and property (P4) if ρ is a pure algebraic attribute (it does not depend on the values of u).*

Proof: (P1) and (P2) are straightforward because the 5th step consists in computing a max-tree with hole-filled components on a scalar image ω. The shapes of the hole-filled maxtree being a subset of the shapes of the ToS of ω, it follows that any two shapes are either nested or disjoint.
(P3) With the same proof as in prop. 2, we show that $\forall A \in \mathcal{S}$,



(a) Original image. (b) ω map computed with D_1. (c) ω map computed with D_2. (d) ω map computed with D_3.

Fig. 9. Distance maps ω computed with D_1, D_2, and D_3 on a practical example; distances are depicted through a "heat" look-up table for a better understanding.

Image	Size	#nodes	Avg. depth	Max depth
airplane	262k	81k / 129k	75 / 78	234 / 197
baboon	240k	89k / 129k	41 / 63	95 / 127
barbara	414k	141k / 228k	64 / 136	203 / 306
boats	453k	119k / 208k	83 / 163	232 / 295
goldhill	414k	123k / 240k	58 / 105	223 / 295
house	65k	22k / 35k	36 / 77	154 / 175
lenna	262k	69k / 161k	40 / 59	148 / 193
pepper	262k	102k / 200k	43 / 117	148 / 291

TABLE I. Tree statistics comparison on well-known test images between the ToS on the gray-level image (left side of the columns) and the MToS on the color image (right side of the columns).

if $\forall B \in \mathcal{S}$, $A \cap B = \emptyset$ or $A \subseteq B$ or $B \subseteq A$, then $A \in \mathbb{C}$ and $A \in \boldsymbol{S}$.
(P4) A marginal tree $\mathcal{T}_i$ only depends on the i^{th} channel, thus it is invariant w.r.t. to u_j ($j \neq i$). By property of the ToS, $\mathcal{T}_i$ is contrast change/inversion invariant w.r.t. u_i. It follows that every $\mathcal{T}_i$ is marginally invariant w.r.t. u and so does the set of shapes $\mathcal{S}$ to build the graph $\mathcal{G}$. Since the rest of the process only depends on the graph topology and no more on the values of u, $\mathcal{T}_\omega$ is thus marginally contrast change/inversion invariant. ■

Proposition 4. *The method provides a tree $\mathcal{T}_\omega$ which is well-formed (P0).*

To verify the property (P0), we have computed some statistics about the tree topology on some classical images. In Table I, we show the number of nodes, the average node depth and the height of the MToS compared to the ToS computed on the gray level version of the image. The number of nodes in the MToS is 50% to 100% higher than in the ToS that highlights a better precision. Moreover, the average depth of the nodes (as well as the height of the tree) increases significantly meaning that we do not just add some leaf nodes (noise) but rather, some large shapes that increase the shape inclusion chains.

V. Related Works

The definitions of level lines (in terms of iso-level sets and as contour of shapes) given in Section II-A are both ill-formed when dealing with partial orders. Indeed, iso-level sets in multivariate images do not form closed curves and the shapes issued from lower and uppers level sets may intersect without being nested.

An unacceptable but widely used workaround for color image processing is to get rid of the colors and process a gray-level version of the multivariate image. This workaround makes sense if we pretend that the geometric information is mainly held by the luminance [30]. However, it is not that rare to face images where edges are only available in the color space (especially document and synthetic images). They contradict this assumption and prove that the chrominance holds the geometric information as well (see Fig. 10b).

Another commonly used solution is processing the image channel-wise and finally recombine the results. Marginal processing is subject to the well-known false color problem as it may create new values that were not in the original images. False colors may or may not be a problem in itself (*e.g.* if the false colors are perceptually close to the original ones) but for image simplification it may produce undesirable artifacts as shown in Fig. 10c. Also marginal processing lead to several trees (each of them representing a single channel the image), whereas we aim at producing a single representation of the image. In [31], we proposed some preliminary ideas to build a single tree from several trees.

Since the pitfall of overlapping shapes is due to the partial ordering of multivariate data, some authors tend to impose an "arbitrary" total ordering or total pre-ordering on values. They differ in the fact that a node may get associated with several values. The way of ordering a multivariate space has been widely studied to extend gray-scale morphological operators. Barnett [32] classifies the methods in four groups: marginal ordering (M-ordering), conditional ordering (C-ordering), partial ordering (P-ordering), and reduced ordering (R-ordering). While the first class leads to partial ordering, the other three ones yield either a total order or a total preorder. C-orderings aim at ordering the vectors giving priorities to some (or all) of the vector components. The well-known lexicographical ordering belongs to this class and is the most common approach to extend the ToS on color [33]. R-ordering methods aim at projecting vectorial values to a scalar through a ranking function. Commonly used ranking functions are for example the l_1-norm, the luminance in a given color space or the distance to a reference set of values. They have been applied by Tushabe and Wilkinson [27], Perret et al. [34], Naegel and Passat [26] for image compression or astronomical object

(a) Original image u.

(b) Simplification on a gray-level version of u (198 regions).

(c) Simplification with a marginal processing (123 + 141 + 136 regions).

(d) Simplification with a non-local rank transformation [29] (total order) (171 regions).

(e) Simplification with our approach (158 regions).

Fig. 10. Simplification issues with "standard" color image processing. (b) shows the leakage problem when the luminance is not sufficient to retrieve the whole geometric information. (c) shows the false color problem due to marginal processing. (e) The method we propose retrieves correctly the main content of the image while preventing false colors.

detection using min or max-trees but the same idea holds for the ToS [35]. More advanced strategies have been designed to build a more "sensitive" total ordering that depends on the image content. Velasco-Forero and Angulo [36, 37] use machine learning techniques to build a P-ordering based on space partitioning and a distance to clusters. In [38], manifold learning is used to infer a ranking function of values and in [39] a locally-dependent ordering are computed on spatial windows. [29] combines both ideas for a manifold learning in a domain-value space capturing small dependencies between a pixel and its neighbors during the construction of the total order. This is illustrated on Fig. 10d. A non-exhaustive list of methods to infer a total order on multivariate data can be found in Aptoula and Lefèvre [40].

Another approach introduced by Passat and Naegel [24] uses directly the partial ordering of values and manipulates the underlying structure that is a graph. The component-graph is still at a development level but has shown promising results for filtering tasks [41]. However, the component-graph faces an algorithmic complexity issue that compels the authors to perform the filtering locally. Thus, the component-graph is currently not suitable if we want to have a single representation of the whole image.

In [31], we introduced a new approach where, instead of trying to impose a total ordering on values, we compute marginally the ToS's and merge them into a single tree. The merge decision does not rely on values anymore but rather on some properties computed in a shape space. However, the merging procedure proposed in that paper shows a loss of "coherence" by merging unrelated shapes together. In [35], inspired by the work of Passat and Naegel [24, 25], we proposed the graph of shapes which merges the marginal ToS's into a single structure in an efficient way. We showed that this structure has a strong potential compared to the standard methods that impose a total order on values. Yet, the method builds a graph that is a limitation since we cannot use the methods provided by the component tree framework (filtering, object detection, segmentation methods...) The work presented in this paper can be seen as a continuation of the ideas introduced in [31] and [35] since the GoS is used as an intermediate representation to extract a single tree from the shapes given by the marginal ToS's.

Finally, it should be noted that there exists other kind of morphological hierarchical representation such as binary partition trees [42], quasi-flat zones hierarchies [43], watershed trees [44, 23]. Those are out-of-the scope of the current paper because they have a different semantic from morphological trees (min/max-trees and ToS). First, they are hierarchies of segmentations where any cut leads to a partition of the image. In addition, in such hierarchies, a node/region is the union of its children nodes/regions; it is not the case for morphological trees, where a node exists precisely because it has some own ("proper") pixels. Second, they belong to the hierarchical clustering algorithms class where the fundamental idea is to cluster similar regions that are adjacent. Thus, they rely on a region similarity criterion (typically a distance involving the gradient) and the adjacency information, whereas morphological trees involve the inclusion of components and the ordering of values. Moreover, it is not worthless to mention that morphological trees are contrast change invariant w.r.t. to the *original* image values, whereas the others are not.

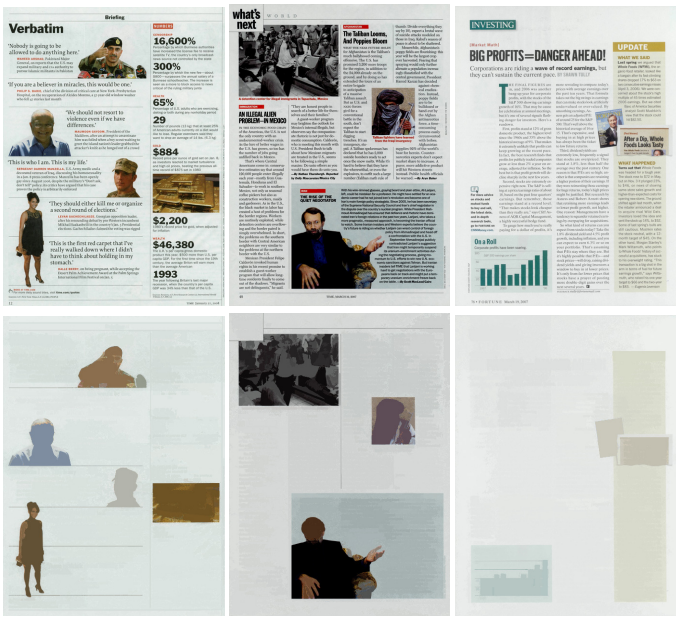


Fig. 11. Simple filtering for document layout detection. Top row: original images; bottom row: results of grain filters.

VI. Illustrations

In this section, we show how versatile is the MToS and that it is well adapted to many differents applications including document and natural images analysis, satellite imaging, medical imaging…

A. Image Filtering

a) Grain Filters in Document Layout Extraction: A grain filter [8, 45] is an operator that removes the regions of the image which are local extrema and whose area is below a given threshold. Using the ToS, a grain filter is thus simply a pruning removing the nodes which do not pass the size criterion. In [21], the grain filter has been used to reveal the "correctness" of the tree in the sense that a small grain size should filter out what we perceive as noise or details while an high grain size should show the main objects and the structure of the image. Here, we use a grain filter to extract text boxes and graphical parts of documents. Indeed, text parts are composed of letters which are supposed to be small components if the MToS is well-formed. On the contrary, text boxes and graphical contents are large components that should remain after the filtering. Figure 11 shows the extraction of non-textual content where self-duality matters since text may be over a darker or brighter background. As one can see, the filtered images only contain the graphical content and text boxes while actual letters are in the residue. Last, in all our experiments we do not observe any significant differences between applying a grain filter on the ToS computed either from the RGB space or from the La*b* space; we are thus inclined to think that the ToS is well stable w.r.t. the choice of a color space.

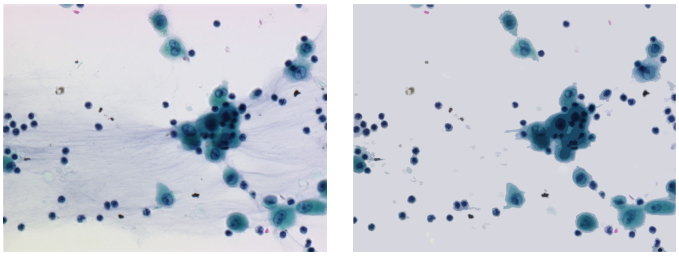


Fig. 12. Microscopic image simplification with the MToS using a *shaping* to filter out non-circular objects; from the image (left, [46]), only 650 shapes remain (right).

b) Shapings for Cytology: In this assessment, we aim at simplifying an image by filtering out objects that do not have a given shape in the context of bronchial cytology. We rely on *shapings* introduced in [47]. We first valuate a two-term energy $E = E_1 + E_2$ on the tree. E_1 expresses the circularity of the shape S: $E_1(S) = 1 - \lambda_2/\lambda_1$ where λ_1 and λ_2 are respectively the lengths of the semi-major axis and semi-minor axes of the best fitting ellipse. The second term E_2 expresses the compacity of the shape: $E_2(S) = \text{Perimeter}(S)^2/\text{Area}(S)$.

Then, we look for the shapes that minimize E. Because the energy varies slightly along a branch, we cannot just threshold the energy as it would preserve many close shapes. *Shapings* solve this problem by building a second hierarchy over the MToS, (*i.e.*, according to the tree topology and the energy) and preserve only the shapes that are local minima. We can further simplify the image by computing the extinction values of the minima and filter out those non-meaningful. Figure 12 shows the simplification on a bronchial microscopical image. As one can see, only nucleus and cytoplasm are well-preserved and the background correctly removed. Such a simplified image can then be combined to a classifier to improve the classification accuracy.

B. Image Simplification

To illustrate the ability of the MToS to represent the main structures of the images, we tested the tree against image simplification. A first assessment uses the method proposed by Xu et al. [48] that minimizes the Mumford-Shah cartoon model constrained by the tree topology. More formally, we have to select a subset of shapes $\mathcal{S}' \subset \mathcal{S}$ that minimizes the energy:

$$E(\mathcal{S}') = \sum_{S \in \mathcal{S}'} \sum_{x \in S | S_x = S} ||f(x) - \bar{f}(S)||_2^2 \; + \; \lambda \, |\partial S|, \qquad (6)$$

where S_x denotes the smallest shape containing x, $\bar{f}(S)$ is the average color of the region and $|\partial S|$ the length of the shape boundary. In [48], the authors use a greedy algorithm that removes the level lines sorted by meaningfulness until the energy does not decrease anymore.

Figure 13 shows the simplification on natural images. While dividing the number of level lines by about 200, the main geometric information is preserved by the simplification. Also, those images are typical cases where the chrominance plays an

Fig. 13. Natural image simplification with the MToS. Left: original images; Right: the simplification running on the MToS. The same λ parameter ($\lambda = 5000$) is used for both images; the simplified images have less than 100 level lines.

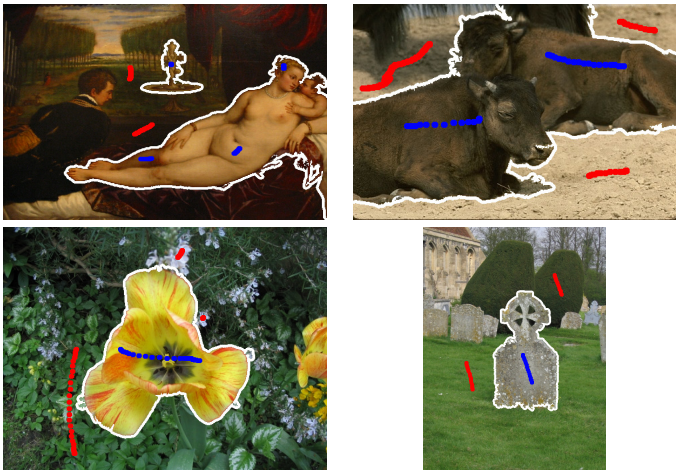
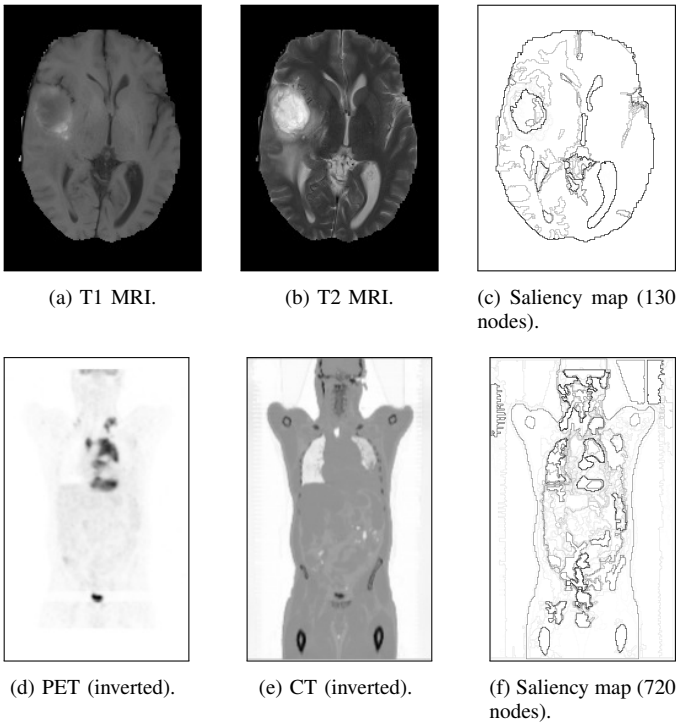


Fig. 14. Object picking with our method. Red and blue user scribbles define the background $\mathcal{B}$ and the foreground $\mathcal{F}$ respectively. The white line is the computed $\mathcal{F}/\mathcal{B}$ boundary.

important role to distinguish regions with similar brightness. Low-contrasted boundaries between regions of similar luminance create a "leakage" effect and level lines merge unrelated objects that are disjoints with the MToS.

C. Interactive Object Segmentation

Inspired by the previous work of [28], in [21] we have introduced a method for interactive image segmentation using the MToS. Given a set of markers B and F (both in $\mathcal{P}(E)$), where B stands for the background class $\mathcal{B}$ and F for the foreground class $\mathcal{F}$, we aim at classifying the other pixels to



(a) T1 MRI. (b) T2 MRI. (c) Saliency map (130 nodes).

(d) PET (inverted). (e) CT (inverted). (f) Saliency map (720 nodes).

Fig. 15. Saliency map on multimodal medical images. The result is a hierarchy of segmentations, in which the objects of interest can be searched for.

one of these classes. [28] use the Nearest Neighbor classifier where the distance between two points x and y is the minimal total variation along all the paths from x to y (see Eq. (4)). The ToS allows a fast computation of the distance between any two points x and y by summing up the variations along the paths of S_x and S_y to their least common ancestors. As a consequence, instead of working at the pixel level, the classification can be done equivalently on the ToS by computing the influence zones of the shapes having a marker pixel using the tree topology. However, our approach differs from [28] as they compute the ToS on a 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. Here, we compute directly the MToS on the original image. To estimate the distance on the tree between a shape and its parent, we simply consider the L^2-distance between their average colors (in RGB, or better in the La*b* space).

The method is illustrated on Fig. 14. A strong advantage is its ability to recover large regions of interest with very few markers, whereas many other methods using statistics require larger markers for a better learning accuracy.

D. Hierarchical Segmentation on Multimodal Images

In [47], the authors proposed to use the ToS in order to get a hierarchical segmentation of the image. Given an attribute (an

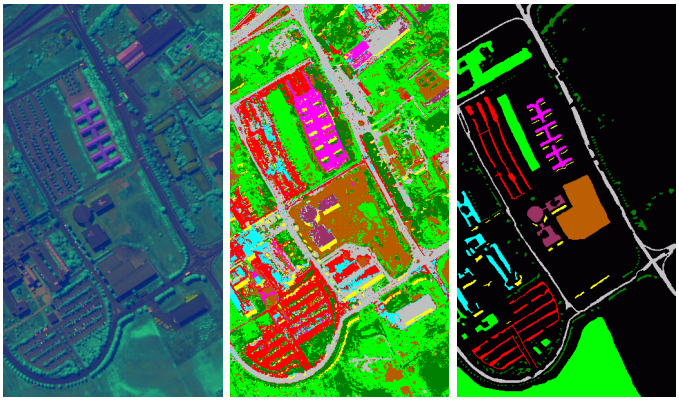


Fig. 16. Hyperspectral image classification. (a) Three principal components recomposed as an RGB image. (b) Classification with the VSDAP using the MToS. (c) Ground truth.

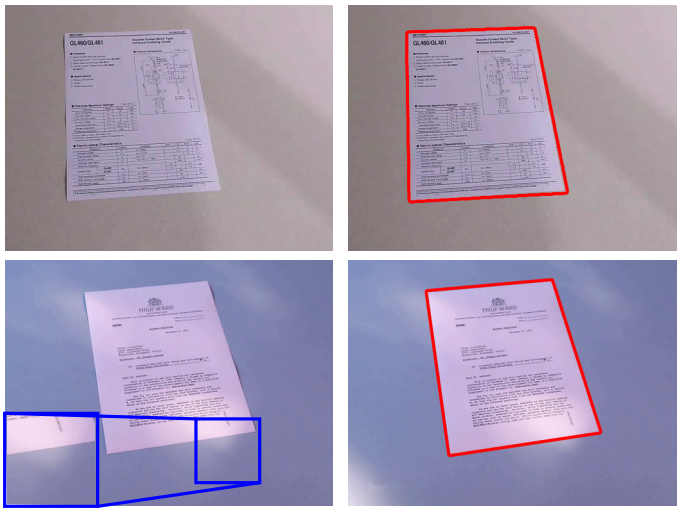


Fig. 17. ICDAR competition on document detection. These images show the robustness of our method to blur and light specularity that move object boundaries. Note that some videos are available as supplementary materials [22].

energy) computed over a tree, one can remove the nodes below a given threshold; it yields a simplification of the image. Yet, there is still a redundancy in the remaining shapes since two close nodes are likely to have the same energy, so to pass the criterion. The idea introduced in [47] is to preserve only the local extrema in the tree and compute their extinction value (the filtering value required such that an extremum gets merged with another extremum). Once the extinction values computed for each extremum node, they are set back on the contour of the shapes in the original domain space and yield a saliency map which is a common way to represent hierarchical segmentation. In other words, thresholding the saliency map with increasing values yields fine to coarse image partitions. This is illustrated in Fig. 15, where images have been acquired with different devices (the brain image with T1- and T2- MRI scans and the body image with PET and CT scans). We have computed the MToS on these images and used the MSER criterion as the energy. As a first remark, the MToS preserves the geometric information of the two channels and mix them in a sensible way. For example, the heart only appears on the PET scan (Fig. 15d) and the lungs in the CT scan (Fig. 15e) but both appear on saliency map. Second, as one can see, the most important objects appear with an high saliency so they will be filter last on the segmentation hierarchy.

E. Classification in Hyperspectral Image

Dalla Mura et al. [49] proposed to use morphological attribute profiles to perform the classification of hyperspectral images acquired by Quickbird. The basic idea is to study pixel-wise the behavior of an operator at different strength of filtering. Because no assumption can be made about the type (dark or light) of the objects to detect, they were interested in self-dual profiles. As a consequence, in [49], they compute a set of attribute openings and closings channel-wise at different predefined thresholds that yields the feature space on which they perform the classification. More formally, given an attribute opening γ_λ and the attribute closing ϕ_λ where $\lambda \in \{\lambda_1, \ldots, \lambda_n\}$ is a family of threshold values, they define their feature map ω as: $\omega(x) = \{\, \gamma_{\lambda_i}(u_k)(x),\, \phi_{\lambda_i}(u_k)(x) \,\}_{k,\,\lambda_i}$.

In [49], they replace the min and max-trees used for the computation of the openings and the closings by self-dual attribute filters using the ToS. They further compare the results in [12] and show that the classification with self-dual attribute profiles outperforms the previous approach with min and max-trees. Yet, the ToS is computed marginally on each channel independently, so we extend their approach with the MToS. The experiments were carried out on a hyperspectral image acquired on Pavia (Italy) by Quickbird. Trees are computed on the first three components of the Principal Component Analysis. The Fig. 16 shows the results of the classification with attributes profiles (AP), the marginal self-dual attribute profiles (MSDAP) and the vectorial self-dual attribute profiles (VSDAP). For each method, the same attribute (moment of inertia), the same filtering parameters (0.1, 0.2, ..., 1.0), and the same classifier (Random Forest) are used. The VSDAP gets an Overall Accuracy of 82.2% while in the same time the AP and MSDAP achieve respectively 77.5% and 68.9%. Using the MToS for filtering and computing the profiles clearly outperforms the ones based on the other trees and tends to confirm that the MToS retrieves and synthesizes correctly the geometric information available in the different channels.

F. Object Detection in Videos

In the scope of the ICDAR competition on Smartphone Document Capture and OCR (SmartDoc-2015) [50], we aim at automatically detecting documents in video captured by smartphones. The dataset covers different document layout (textual and/or having graphical content) and realistic scene analysis problems (change of illumination, motion blur, change of perspectives, partial occlusions...). Yet, the method we propose relies on the MToS representation of the image. Basically, we aim at identifying some nodes in the tree that match some document criteria expressed as shape attributes

and as an energy. The two criteria are: 1. How much the shape boundary fits a quadrilateral, *i.e.*, for each shape A we compute the best fitting quadrilateral Q_A and we measure the ratio: $E_1(A) = |A|/|Q_A|$. 2. How "noisy" is the object (we expect a document with some text and graphics *i.e.*, a shape that contains many shapes inside): $E_2(A) = (\sum_{X \in \mathcal{S},\, X \subset A} d(X))/|A|$, where $d(X)$ stands for the depth of every shape X included in A.

We then look for the shape that minimizes the energy $E_1(X) + E_2(X)$. Note that for a better accuracy of E_2, we start with preprocessing the image with a grain filter to reduce the effect of the natural image noise.

The method (slightly modified to enable document tracking between frames) got the first place of the competition among 7 participants. The evaluation was based on the Jaccard index, that measures the similarity between the set of expected pixels in the ground truth and the set of the segmentation result returned by the method. Our method obtained an average Jaccard index of 0.9716, varying between 0.9710 and 0.9721 on the whole dataset [50], which tends to show the robustness of the proposed approach.

VII. Conclusion

In this paper, we have presented a method to extend the morphological structure of Tree of Shapes to multivariate images. Contrary to classical approaches, the Multivariate Tree of Shapes (MToS) does not rely on any arbitrary ordering of multivariate data; it is only based on the inclusion relationship between shapes. We have also investigated the structural properties of the MToS, and we have proved the validity of its definition regarding these properties. In particular, the resulting tree is marginally both self-dual and invariant to contrast changes. Eventually, we have shown the versatility and the potential of this tree-based representation through a wide range of illustrations involving different kinds of multivariate data: color images and videos, hyperspectral data, and multimodal medical images. The most prominent feature of having such a tree-based representation is the simplicity of processing: filtering and simplifying images is performed by removing appropriate nodes, selecting interactively some objects by labeling nodes, and so on. The perspectives of our work is to show that the MToS representation is competitive when considering practical applications; so we will quantitatively compare the results we can obtain by processing the MToS with those of some state-of-art methods. As a matter of reproducible research, the binaries of the MToS-based illustrations presented in this paper are available online [22]. In addition, the source code of the tree construction is also available. It is based on our free software image processing library [51], and we intend to include it in the next release of our library.

Acknowledgments

The authors would like to warmly thank M. Dalla Mura for helping us on hyperspectral image classification, O. Lézoray for providing the rank images in Fig. 10 (d), and the "Smart-Doc competition" team for having organized this challenge (and especially to Joseph Chazalon for having pointed out to us this challenge).

References

[1] V. Caselles, B. Coll, and J.-M. Morel, "Topographic maps and local contrast changes in natural images," *Intl. J. 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 Trans. on Image Processing*, vol. 9, no. 5, pp. 860–872, 2000.

[3] T. Géraud, E. Carlinet, S. Crozet, and L. Najman, "A quasi-linear algorithm to compute the tree of shapes of n-D images." in *Proc. of Intl. Symp. on Mathematical Morphology (ISMM)*, ser. LNCS, vol. 7883. Uppsala, Sweden: Springer, 2013, pp. 98–110.

[4] S. Crozet and T. Géraud, "A first parallel algorithm to compute the morphological tree of shapes of nD images," in *Proc. of IEEE Intl. Conf. on Image Processing (ICIP)*, Paris, France, 2014, pp. 2933–2937.

[5] E. Carlinet and T. Géraud, "A comparative review of component tree computation algorithms," *IEEE Trans. on Image Processing*, vol. 23, no. 9, pp. 3885–3895, 2014.

[6] Y. Xu, E. Carlinet, T. Géraud, and L. Najman, "Efficient computation of attributes and saliency maps on tree-based image representations," in *Proc. of Intl. Symp. on Mathematical Morphology (ISMM)*, ser. LNCS, vol. 9082. Reykjavik, Iceland: Springer, 2015, pp. 693–704.

[7] F. Dibos and G. Koepfler, "Total variation minimization by the Fast Level Sets Transform," in *IEEE Workshop on Variational and Level Set Methods in Computer Vision*, 2001, pp. 179–185.

[8] V. Caselles and P. Monasse, "Grain filters," *Journal of Mathematic Imaging and Vision*, vol. 17, no. 3, pp. 249–270, Nov. 2002.

[9] Y. Xu, T. Géraud, and L. Najman, "Morphological filtering in shape spaces: Applications using tree-based image representations," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, Tsukuba Science City, Japan, 2012, pp. 485–488.

[10] Y. Xu, T. Géraud, P. Monasse, 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.

[11] G.-S. Xia, J. Delon, and Y. Gousseau, "Shape-based invariant texture indexing," *Intl. J. of Computer Vision*, vol. 88, no. 3, pp. 382–403, 2010.

[12] G. Cavallaro, M. Dalla Mura, J. A. Benediktsson, and L. Bruzzone, "A comparison of self-dual attribute profiles based on different filter rules for classification," in *Proc. of the IEEE Intl. Geoscience and Remote Sensing Symp. (IGARSS)*, Jul. 2014, pp. 1265–1268.

[13] Y. Xu, T. Géraud, and L. Najman, "Context-based energy estimator: Application to object segmentation on the tree of shapes," in *Proc. of IEEE Intl. Conf. on Image Processing (ICIP)*, Orlando, USA, 2012, pp. 1577–1580.

[14] N. Widynski, T. Géraud, and D. Garcia, "Speckle spot detection in ultrasound images: Application to speckle reduction and speckle tracking," in *Proc. of the IEEE Intl. Ultrasonics Symposium (IUS)*, Chicago, IL, USA, 2014, pp. 1734–1737.

[15] V. Caselles and P. Monasse, *Geometric Description of Images as Topographic Maps*, ser. Lecture Notes in Mathematics. Springer-Verlag, 2009, vol. 1984.
[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] P. Salembier and J. Serra, "Flat zones filtering, connected operators, and filters by reconstruction," *IEEE Trans. on Image Processing*, vol. 4, no. 8, pp. 1153–1160, 1995.
[18] H. J. A. M. Heijmans, "Self-dual morphological operators and filters," *Journal of Mathematic Imaging and Vision*, vol. 6, no. 1, pp. 15–36, 1996.
[19] P. Soille, "Beyond self-duality in morphological image analysis," *Image and Vision Computing*, vol. 23, no. 2, pp. 249–257, 2005.
[20] A. Desolneux, L. Moisan, and J. Morel, "Edge detection by Helmholtz principle," *Journal of Mathematic Imaging and Vision*, vol. 14, no. 3, pp. 271–284, 2001.
[21] E. Carlinet and T. Géraud, "A color tree of shapes with illustrations on filtering, simplification and segmentation," in *Proc. of Intl. Symp. on Mathematical Morphology (ISMM)*, ser. LNCS, vol. 9082. Reykjavik, Iceland: Springer, 2015, pp. 363–374.
[22] E. Carlinet, "Supplementary materials for this paper (MToS: A tree of shapes for multivariate images)," 2015. [Online]. Available: http://publications.lrde.epita.fr/carlinet.15.itip
[23] L. Najman and T. Géraud, "Discrete set-valued continuity and interpolation," in *Proc. of Intl. Symp. on Mathematical Morphology (ISMM)*, ser. LNCS, vol. 7883. Uppsala, Sweden: Springer, 2013, pp. 37–48.
[24] N. Passat and B. Naegel, "An extension of component-trees to partial orders," in *Proc. of IEEE Intl. Conf. on Image Processing (ICIP)*, 2009, pp. 3933–3936.
[25] ——, "Component-trees and multivalued images: Structural properties," *J. of Mathematic Imaging and Vision*, vol. 49, no. 1, pp. 37–50, 2014.
[26] B. Naegel and N. Passat, "Component-trees and multivalue images: A comparative study," in *Proc. of Intl. Symp. on Mathematical Morphology (ISMM)*, ser. LNCS, vol. 5720. Springer, 2009, pp. 261–271.
[27] F. Tushabe and M. H. F. Wilkinson, "Color processing using max-trees: A comparison on image compression," in *Proc. of International Conference on Systems and Informatics (ICSAI)*. IEEE, 2012, pp. 1374–1380.
[28] A. Dubrovina, R. Hershkovitz, and R. Kimmel, "Image editing using level set trees," in *Proc. of IEEE Intl. Conf. on Image Processing (ICIP)*, 2014, pp. 4442–4446.
[29] O. Lézoray and A. Elmoataz, "Nonlocal and multivariate mathematical morphology," in *Proc. of IEEE ICIP*, Orlando, USA, 2012, pp. 129–132.
[30] V. Caselles, B. Coll, and J.-M. Morel, "Geometry and color in natural images," *Journal of Mathematic Imaging and Vision*, vol. 16, no. 2, pp. 89–105, 2002.
[31] E. Carlinet and T. Géraud, "A morphological tree of shapes for color images," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, Stockholm, Sweden, Aug. 2014, pp. 1132–1137.
[32] V. Barnett, "The ordering of multivariate data," *Journal of the Royal Statistical Society. Series A (General)*, vol. 139, no. 3, pp. 318–355, 1976.
[33] B. Coll and J. Froment, "Topographic maps of color images," in *Proc. of International Conference on Pattern Recognition*, vol. 3, 2000, pp. 609–612.
[34] B. Perret, S. Lefèvre, C. Collet, and E. Slezak, "Connected component trees for multivariate image processing and applications in astronomy," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, Aug. 2010, pp. 4089–4092.
[35] E. Carlinet and T. Géraud, "Getting a morphological tree of shapes for multivariate images: Paths, traps and pitfalls," in *Proc. of IEEE Intl. Conf. on Image Processing (ICIP)*, Paris, France, 2014, pp. 615–619.
[36] 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.
[37] ——, "Random projection depth for multivariate mathematical morphology," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 7, pp. 753–763, 2012.
[38] O. Lézoray, 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.
[39] O. Lézoray, C. Meurie, and A. Elmoataz, "A graph approach to color mathematical morphology," in *Proc. IEEE Intl. Symp. on Signal Processing and Information Technology*, 2005, pp. 856–861.
[40] E. Aptoula and S. Lefèvre, "A comparative study on multivariate mathematical morphology," *Pattern Recognition*, vol. 40, no. 11, pp. 2914–2929, 2007.
[41] B. Naegel and N. Passat, "Towards connected filtering based on component-graphs," in *Proc. of Intl. Symp. on Mathematical Morphology (ISMM)*, ser. LNCS, vol. 7883. Springer, 2013, pp. 353–364.
[42] 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.
[43] 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.
[44] S. Beucher and F. Meyer, "The morphological approach to segmentation: the watershed transformation," in *Mathematical Morphology in Image Processing*, E. R. Dougherty, Ed., New York, 1993, pp. 433–481.
[45] N. Ray and S. T. Acton, "Inclusion filters: A class of self-dual connected operators," *IEEE Transactions on Image Processing*, vol. 14, no. 11, pp. 1736–1746, 2005.
[46] C. Meurie, O. Lézoray, C. Charrier, and A. Elmoataz, "Combination of multiple pixel classifiers for microscopic image segmentation," *IASTED Intl. J. of Robotics and Automation*, vol. 20, no. 2, pp. 63–69, 2005.
[47] Y. Xu, T. Géraud, and L. Najman, "Two applications of shape-based morphology: Blood vessels segmentation

and a generalization of constrained connectivity," in *Proc. of Intl. Symp. on Mathematical Morphology (ISMM)*, ser. LNCS, vol. 7883. Springer, 2013, pp. 390–401.

[48] ——, "Salient level lines selection using the Mumford-Shah functional," in *Proc. of IEEE Intl. Conf. on Image Processing (ICIP)*, Melbourne, Australia, 2013, pp. 1227–1231.

[49] M. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone, "Morphological attribute profiles for the analysis of very high resolution images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 10, pp. 3747–3762, 2010.

[50] J.-C. Burie, J. Chazalon, M. Coustaty, S. Eskenazi, M. M. Luqman, M. Mehri, N. Nayef, J.-M. Ogier, S. Prum, and M. Rusiol, "SmartDoc: Smartphone document capture and OCR competition," in *Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR'15)*. Nancy, France: IEEE, 2015, to appear.

[51] 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 Intl. Conf. on Image Processing (ICIP)*, Hong Kong, 2010, pp. 1941–1944.

Edwin Carlinet received the Ing. degree from EPITA, Paris, France, in 2011, and a M.Sc. in applied mathematics for computer vision and machine learning from the École Normale Supérieure Cachan, in 2012. He is currently a Ph.D. candidate with EPITA and Université Paris-Est. His research interests include bio-informatics mathematical morphology, and statistical learning.

Thierry Géraud received a Ph.D. degree in signal and image processing from Télécom ParisTech in 1997, and the Habilitation à Diriger les Recherches from Université Paris-Est in 2012. He is one of the main authors of the Olena platform, dedicated to image processing and available as free software under the GPL licence. His research interests include image processing, pattern recognition, software engineering, and object-oriented scientific computing. He is currently working at EPITA Research and Development Laboratory (LRDE), Paris, France.