A Color Tree of Shapes with Illustrations on Filtering, Simplification, and Segmentation

Edwin Carlinet[1][2] and Thierry Géraud[1]

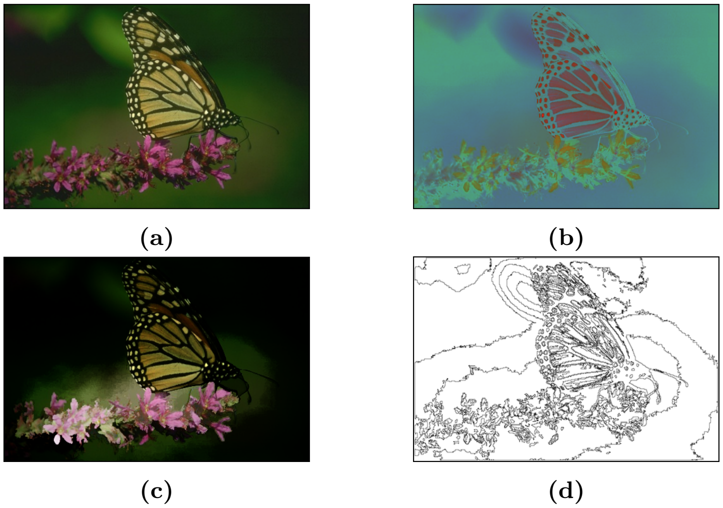
[1] EPITA Research and Development Laboratory (LRDE)
{firstname.lastname}@lrde.epita.fr
[2] Université Paris-Est, LIGM, Équipe A3SI, ESIEE

Abstract. The Tree of Shapes (ToS) is a morphological tree that provides a high-level, hierarchical, self-dual, and contrast invariant representation of images, suitable for many image processing tasks. When dealing with color images, one cannot use the ToS because its definition is ill-formed on multivariate data. Common workarounds such as marginal processing, or imposing a total order on data are not satisfactory and yield many problems (color artifacts, loss of invariances, etc.) In this paper, we highlight the need for a self-dual and contrast invariant representation of color images and we provide a method that builds a single ToS by merging the shapes computed marginally, while guarantying the most important properties of the ToS. This method does not try to impose an arbitrary total ordering on values but uses only the inclusion relationship between shapes. Eventually, we show the relevance of our method and our structure through some illustrations on filtering, image simplification, and interactive segmentation.

Keywords: tree of shapes, hierarchical representation, color image processing

1 Introduction

The Tree of Shapes (ToS) [6] is a hierarchical representation of an image in terms of the inclusion of its level lines. Its powerfulness lies in the number of properties verified by this structure. First, it is a morphological representation based on the inclusion of the connected components of an image at the different levels of thresholding. As such, a basic filtering of this tree is a connected filter, i.e., an operator that does not move the contours of the objects but either keep or remove some of them [16]. In addition, not only is it invariant by contrast changes on the whole image but it is also robust to local changes of contrast [2]. This property is very desirable in many computer vision applications where we face the problem of illumination changes, e.g., for scene matching, object recognition... In Fig. 1c, we show this invariance by simulating a change of illumination directly in the ToS so we do have the exact same tree representation as for the original image in Fig. 1a. Third, besides being contrast change invariant, the ToS is also a self-dual representation of the image. This feature is fundamental in a context where structures may appear both on a brighter


(a)

(b)

(c)

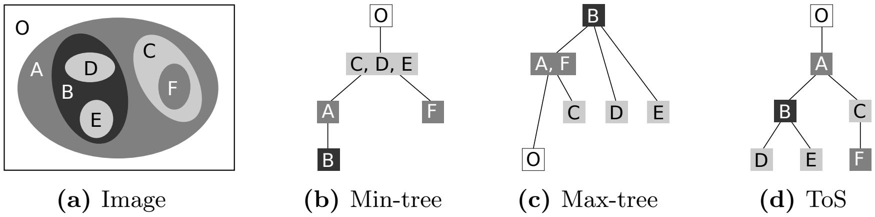
(d)

Fig. 1. On the need for contrast change/inversion invariance. (b) and (c) have been generated from the original image. We have changed and/or inverted the contrast of each channel for (b) and we have changed locally the contrast for (c) (to simulate a change of illumination). The three images (a), (b), and (c) give the same CToS whose level lines are shown in (d).

background or on a lighter one. Therefore, self-dual operators are particularly well adapted to process images without prior knowledge on their contents. While many morphological operators try to be self-dual by combining extensive and anti-extensive operators (for instance, the Alternating Sequential Filters), many of them actually depend on the processing order (i.e., on which filter comes first). Self-dual operators have the ability to deal with both dark and light objects in a *symmetric* way [12,17] (see Fig. 1b).

Despite all these wonderful features, the ToS is still widely under-exploited even if some authors have already been using it successfully for image processing and computer vision applications. In [9,21,22], an energy optimization approach is performed on the ToS for image simplification and image segmentation, while Cao et al. [2] rely on an *a-contrario* approach to select meaningful level-lines. Some other applications include blood vessel segmentation [21], scene matching extending the Maximally Stable Extremal Regions (MSER) through the Maximally Stable Shapes [6], image registration [6], and so on.

While the ToS is well-defined on grayscale images, it is getting more complicated with multivariate data. Indeed, like in the case of the min and max-trees, the ToS relies on an ordering relation of values which has to be total. If it is not, the definition based on lower and upper threshold sets yield components that may overlap and the tree of inclusion does not exist. To overcome this problem, most authors have focused on defining a total order on multivariate data. However, from our point of view, the most important feature of morphological trees lies in the inclusion of components/shapes. As a consequence, this paper introduces a novel approach which does not intend to build a total order, but



(a) Image **(b)** Min-tree **(c)** Max-tree **(d)** ToS

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

tries to build up a set of non-overlapping shapes from an arbitrary set of shapes using the inclusion relation only.

The paper is organized as follows. In Section 2, we remind the basis about the ToS and its formal definition. We also review some classical approaches to handle multivariate data with morphological trees. In Section 3, we introduce a new method to build a Color Tree of Shapes (CToS) by merging the shapes issued from marginal ToS's. In Section 4, we show some illustrations to highlight the potential of the CToS and its versatility.

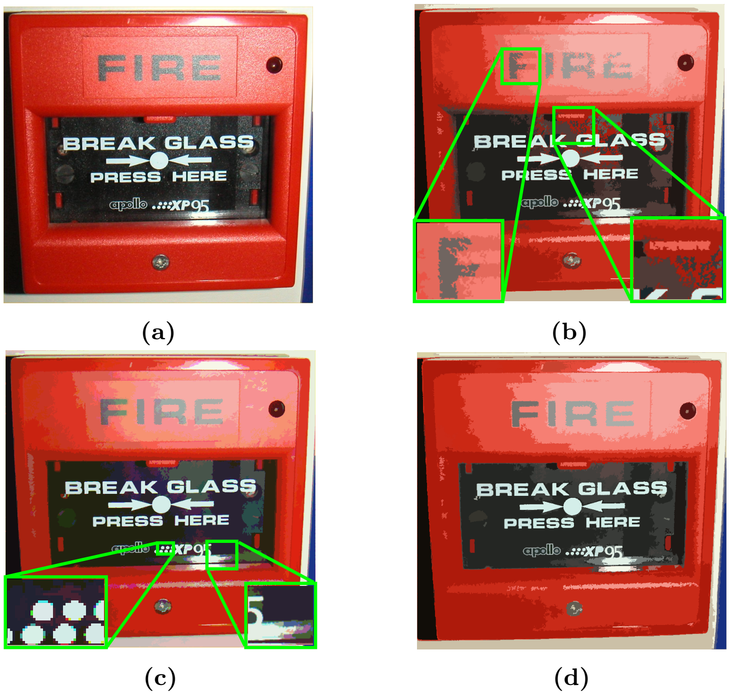
2 Background

2.1 The Tree of Shapes

Let an image $u : \Omega \to E$ defined on a domain Ω and taking 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 cut and upper cut) defined as $[u < \lambda] = \{x \in \Omega,\, u(x) < \lambda\}$. We note $\mathcal{CC}(X), X \in \mathcal{P}(\Omega)$ the set of connected components of X. If $\leq$ is a total relation, any two connected components $X, Y \in \mathcal{CC}([u < \lambda])$ are either disjoint or nested. The set $\mathcal{CC}([u < \lambda])$ endowed with the inclusion relation forms a tree called the *min-tree* and its dual tree, defined on upper cuts, is called the *max-tree* (see Figs. 2b and 2c). Given the hole-filling operator $\mathcal{H}$, we call a *shape* any element of $\mathcal{S} = \{\mathcal{H}(\Gamma),\, \Gamma \in \mathcal{CC}([u < \lambda])\}_\lambda \cup \{\mathcal{H}(\Gamma),\, \Gamma \in \mathcal{CC}([u > \lambda])\}_\lambda$. 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. 2d). 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 in [11], one can ensure each level line is an isolevel closed curve given that $\leq$ is a total order. Actually, the “totality” requirement about $\leq$ comes from the definition of the level lines in terms of contours of lower or upper sets. 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 ($n > 1$).

2.2 The Color Problem: Common Solutions and Related Works

The previous definitions of level lines (in terms of iso-level curves and as contour of shapes) are both ill-formed when dealing with partial orders. Indeed, iso-level



(a) **(b)** **(c)** **(d)**

Fig. 3. Simplification issues with "classical" color image processing. (b) shows the simplification on the luminance of the original image (198 regions) issuing leakage problems because it does not allow to retrieve the whole geometric information. (c) shows the marginal processing (123 + 141 + 136 regions) that introduces false colors. (d) is the simplification with our method (158 regions). It retrieves correctly the main contents of the image while preventing the introduction of false colors.

sets in colors do not form closed curves and the shapes issued from lower and uppers cuts may intersect without being nested, i.e., $(\mathcal{S}, \subseteq)$ is a graph.

An unacceptable but widely used workaround for color image processing is to get rid of colors and to process a gray-level version of the color image. This workaround makes sense if we pretend that the geometric information is mainly held by the luminance [5]. However, many images exist where edges are only available in the color space (e.g. in document or synthetic images), emphazing that the chrominance holds some geometric information as well (see Fig. 3b).

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: it usually creates new colors 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 color artifacts as shown in Fig. 3c. Also marginal processing leads to several trees (each of them representing a single channel the image) whereas we aim at producing a single representation of the image.

Since the pitfall of overlapping shapes is due to the partial ordering of colors, 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 colors. The way of ordering a multivariate space has been widely studied to extend gray-scale morphological operators. A non-extensive review of classical way of ordering values can be found in [1]. Also more advanced strategies have been designed to build a more “sensitive” total ordering that depends on the image contents (see for example [19,20,13,8]).

Another approach introduced by [15] uses directly the partial ordering of values and manipulates the underlying structure, which is a graph. The component-graph is still at a development level but has shown promising results for filtering tasks [14]. However, the component-graph faces an algorithmic complexity issue that compels the authors to perform the filtering locally. Thus, it is currently not suitable if we want to have a single representation of the whole image.

In [4], we introduced an 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 [3], inspired by the work of [15], we proposed the Graph of Shapes (GoS) 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 methods that impose a total order on values. Yet, the method builds a graph that prevents from using tools from the component tree framework (filtering, object detection, segmentation methods, etc.) and complicates the processing. The work presented here can be seen as a continuation of the ideas introduced in [4] and [3] since the GoS is used as an intermediate representation to extract a single tree from the marginal ToS’s.

3 Merging the Trees of Shapes

3.1 Overview and Properties

Let us first relax the definition of a *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 primary 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 $\boldsymbol{S}$ such that any two shapes are either nested or disjoint. We do not impose the constraint $\boldsymbol{S} \subseteq \mathcal{S}$. In other words, we do allow the method to build new shapes that were not in the primary shape set. We note $T : \Omega^{\mathbb{R}^n} \to (\mathcal{P}(\mathcal{P}(\Omega)), \subseteq)$ the process that builds a ToS $(\boldsymbol{S}(\mathbf{u}), \subseteq)$ from an image $\mathbf{u} \in (\mathbb{R}^n)^{\Omega}$.

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

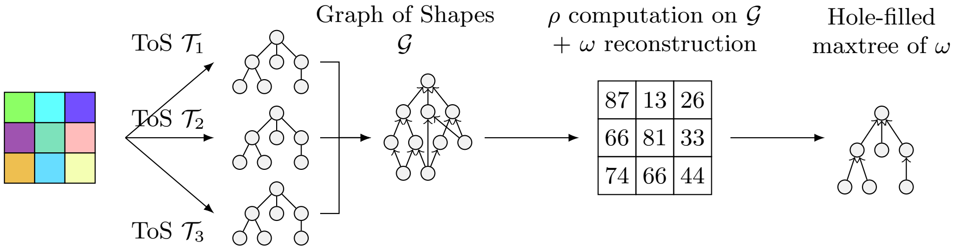


Fig. 4. Scheme of the proposed method to compute a Color Tree of Shapes (CToS).

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 and a representation T is said self-dual and morphological if $T(G(u)) = T(u)$.

More formally, we want the method T to produce $T(\mathbf{u}) = (\boldsymbol{S}(\mathbf{u}), \subseteq)$ having the following properties:

(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) Scalar ToS equivalence. If $\mathcal{M} = \{\mathcal{S}_1\}$ then $\boldsymbol{S}(\mathbf{u}) = \mathcal{S}_1$ (for scalar images, the tree built by the method is equivalent to the gray-level ToS).

(P4) If a shape $X \in \boldsymbol{S}(\mathbf{u})$ verifies:
$$\forall Y \neq X \in \boldsymbol{S},\ X \cap Y = \emptyset \text{ or } X \subset Y \text{ or } Y \subset X$$
then $X \in \boldsymbol{S}(\mathbf{u})$ (any shape that does not overlap with any other shape should exist in the final shape set).

(P5) 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 shall be invariant by marginal inversion/change of contrast, that is, $T(\mathbf{G}(\mathbf{u})) = T(\mathbf{u})$.

3.2 Method Description

The method we propose is a 4-steps process (see Fig. 4). First, we start with computing the marginal ToS's $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n$ of $\mathbf{u}$ associated with the shape sets $\mathcal{S}_1$, $\mathcal{S}_2$, $\ldots \mathcal{S}_n$ that give a primary shape set $\mathcal{S} = \bigcup \mathcal{S}_i$. The multiple trees provide a representation of the original image and $\mathbf{u}$ can be reconstructed marginally from them. However, handling several trees is not straightforward and they lack some important information: how the shapes of one tree are related (w.r.t the inclusion) to the shapes of the other trees. The graph $\mathcal{G}$, the cover of $(S, \subseteq)$, is nothing more than these trees merged in a unique structure that adds the inclusion relation that was missing previously. As a 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, contrast invariant representation of $\mathbf{u}$.

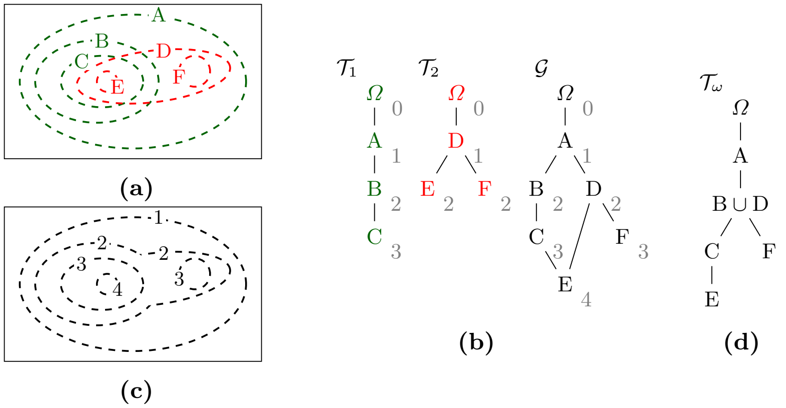
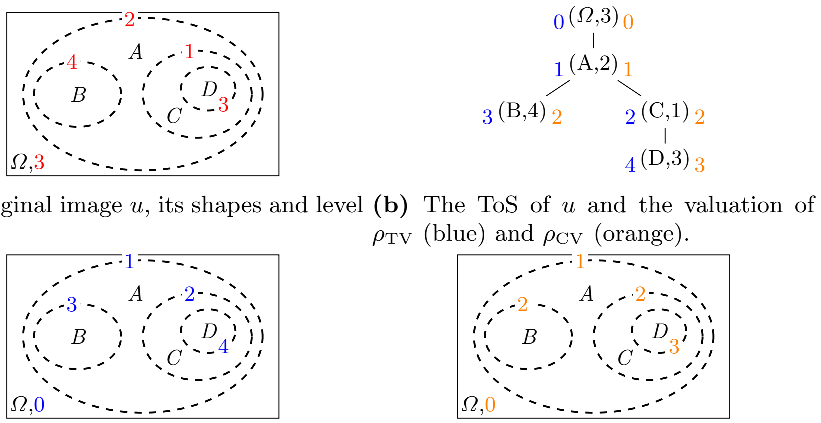


Fig. 5. The method illustrated on an example. (a) A 2-channels image $\mathbf{u}$ and its shapes (resp. in green and red). (b) The marginal ToS's $\mathcal{T}_1$, $\mathcal{T}_2$ and the GoS. The depth appears in gray near the nodes. (c) ω image built from $\mathcal{G}$. (d) The max-tree $\mathcal{T}_\omega$ of ω.

The second part of the method tries to extract a tree from $\mathcal{G}$ verifying the constraints (P1) to (P5). The major difficulty of this task is to get from $\mathcal{G}$ a new set of shapes that do not overlap. The first observation is that for any decreasing attribute $\rho : \mathcal{P}(\Omega) \to \mathbb{R}$ (i.e. $\forall A, B \in \mathcal{S}, A \subset B \Rightarrow \rho(A) > \rho(B)$), 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 on a decreasing attribute. An example of such an attribute is the *depth* where the *depth* of a shape in $\mathcal{G}$ stands for the length of the longest path of a shape A from the root. Consider now the image $\omega(x) = \max_{x \in X, X \in \mathcal{S}} \rho(x)$. ω is an image that associates for each point x, the depth of the deepest shape containing x (see Figs. 5b and 5c). The latter may form component with holes so we consider $\boldsymbol{S}(\mathbf{u}) = \mathcal{H}(\mathbb{C})$ and $(\boldsymbol{S}(\mathbf{u}), \subseteq)$ as the final ToS $\mathcal{T}_\omega$ (see Fig. 5d). Note that in the case where $(\mathcal{S}, \subset)$ is already a tree, we thus have $\mathbb{C} = \{\mathcal{CC}([\omega \geq h]), h \in \mathbb{R}\} = \mathcal{S}$. In other words, the max-tree of w reconstructed from ρ valuated on a tree $\mathcal{T}$ yields the same tree (property **P3**) and more generally, if a shape A do not overlap any other shape, it belongs to $\mathcal{CC}([\omega \geq h])$ (property **P4**). From a computational standpoint, the most expensive part is the graph computation which is $O(n^2.H.N)$, where n is the number of channels to merge, H the maximal depth of the trees, and N the number of pixels. Indeed, for each shape of one tree, we need to track its smallest including shape in the other trees which is basically an incremental least common ancestor attribute computation. The other steps are either linear or quasi-linear with the number of pixels. In the next section, we explain why we choose ρ to be the *depth* in $\mathcal{G}$.

3.3 Computing the Inclusion Map

The 4th step of the method involves the choice of an attribute to be computed over the GoS $\mathcal{G}$. This is a critical step since it decides which shapes are going



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

to be merged or removed. In Section 3.2, the *depth* is used as the attribute to merge the shapes, yet without justification. We now explain the rationale for this choice. Consider the distance measure between two points (p, p') in Ω:

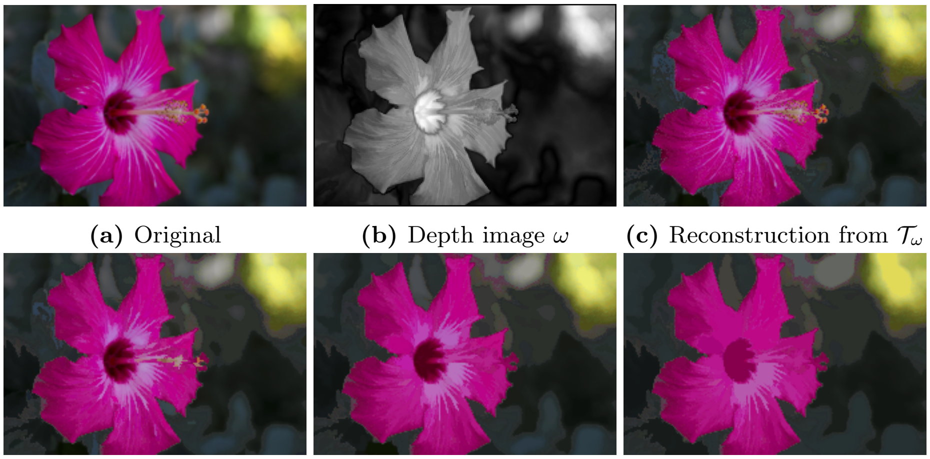
$$d_{\mathrm{TV}}(p, p') = \min_{C_{pp'}} \int_0^1 |\nabla u(C_{pp'}(t)).\dot{C}_{pp'}(t)|.dt, \tag{1}$$

where $C_{pp'}(t)$ is a path in Ω from p to p'. Equation (1) is actually the minimum total variation among the paths from p to p'. This measure has been used by [10] for segmenting where the ToS helps to compute efficiently the level set distance. Let $\omega_{\mathrm{TV}}(x) = d_{\mathrm{TV}}(\partial\Omega, x)$ the Total Variation distance map from the border. It can be computed using 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.

The problem with the Total Variation metric lies in that it depends on u, i.e., ω_{TV} is no contrast invariant. A contrast invariant counterpart would be to only count the number of variation (CV), i.e., the minimum number of level lines to traverse to get to p. Algorithmically speaking, building ω_{CV} consists in computing the depth attribute $\rho_{\mathrm{CV}}(A) = |\{S \in \mathcal{S} \mid A \subset S\}|$ and reconstructing $\omega_{\mathrm{CV}}(x) = \max_{X \in \mathcal{S},\, x \in X} \rho_{\mathrm{CV}}(X)$. This process is shown on Fig. 6. Based on the equivalence between level lines and equidistant lines for scalar images, we would want to build such a distance map for color images as well. Once again, the idea is to count the number of marginal level lines to traverse. More formally:

$$\rho(A) = \max_{\phi \in [\Omega \rightsquigarrow A)} |\phi| \quad \text{and} \quad \omega_{\mathrm{CV}}(x) = \max_{X \in \mathcal{S},\, x \in X} \rho(X)$$

where $[\Omega \rightsquigarrow A)$ stands for the set of paths from the root to A in $\mathcal{G}$. $\omega_{\mathrm{CV}}(x)$ actually counts the number of marginal level lines (*that are nested*) along the path from the border to the deepest shape that contains x. ρ can be computed efficiently from $\mathcal{G}$ using a basic shortest path algorithm.



(a) Original **(b)** Depth image ω **(c)** Reconstruction from $\mathcal{T}_\omega$

(d) Grain filters of sizes 10, 100 and 500

Fig. 7. Grain filters

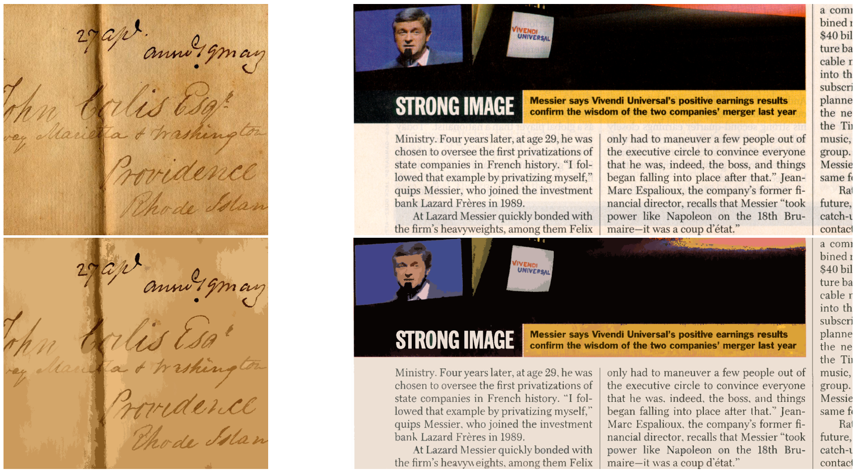
4 Illustrations

4.1 Grain Filters

A grain filter [7] 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. Grain filters allow 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. In Fig. 7, we show the inclusion map ω computed using our method and the image reconstructed from the max-tree $\mathcal{T}_\omega$. The reconstruction consists in computing for each node the average color of the pixels it contains and then, assigning this value to the pixels. Because $\mathcal{T}_\omega$ is not a reversible representation of $\mathbf{u}$, the latter cannot be recovered from $\mathcal{T}_\omega$, however the reconstruction is close to the original. This illustration is rather a “structure validation” experience and does not aim at getting the best filtering results. In particular, we are in the same case of preorder-based filters, where a node may be associated with many color values. More advanced reconstruction strategies can be found in [18] that limit the restitution artifacts. In Fig. 7d, we have applied size-increasing grain filters that eliminate details in a “sensitive” way, and provide a reconstruction with few color artifacts that validate the structure organization of our tree.

4.2 Image Simplification

To illustrate the ability of the CToS to represent the main structures of the images, we tested the tree against image simplification. This assessment uses the method proposed by [22] that minimizes the Mumford-Shah cartoon model



27 apl.
ansd 9 may
John Corlis Esqr.
way Marietta & washington
Providence
Rhode Islan

27 apl.
ansd 9 may
John Corlis Esqr.
way Marietta & washington
Providence
Rhode Islan

VIVENDI UNIVERSAL

STRONG IMAGE Messier says Vivendi Universal's positive earnings results confirm the wisdom of the two companies' merger last year

Ministry. Four years later, at age 29, he was chosen to oversee the first privatizations of state companies in French history. "I followed that example by privatizing myself," quips Messier, who joined the investment bank Lazard Frères in 1989.

At Lazard Messier quickly bonded with the firm's heavyweights, among them Felix only had to maneuver a few people out of the executive circle to convince everyone that he was, indeed, the boss, and things began falling into place after that." Jean-Marc Espalioux, the company's former financial director, recalls that Messier "took power like Napoleon on the 18th Brumaire—it was a coup d'état."

a comı
bined ı
$40 bil
ture ba
cable ı
into th
subscri
planne
the ne
the Ti
music,
group.
Messie
same f
Rat
future,
catch-u
contact

STRONG IMAGE Messier says Vivendi Universal's positive earnings results confirm the wisdom of the two companies' merger last year

Ministry. Four years later, at age 29, he was chosen to oversee the first privatizations of state companies in French history. "I followed that example by privatizing myself," quips Messier, who joined the investment bank Lazard Frères in 1989.

At Lazard Messier quickly bonded with the firm's heavyweights, among them Felix only had to maneuver a few people out of the executive circle to convince everyone that he was, indeed, the boss, and things began falling into place after that." Jean-Marc Espalioux, the company's former financial director, recalls that Messier "took power like Napoleon on the 18th Brumaire—it was a coup d'état."

a comı
bined ı
$40 bil
ture ba
cable ı
into th
subscri
planne
the ne
the Ti
music,
group.
Messie
same f
Rat
future,
catch-u
contact

(a) 112 / 288k level lines selected

(b) 1600 / 120k level lines selected

Fig. 8. Document simplification. Original images on top and simplified images below.

constrained by the tree topology. More formally, we have to select a subset of shapes $\mathcal{S}' \subset \mathcal{S}$ that minimizes the energy:

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

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 [22], the authors use a greedy algorithm that removes the level lines sorted by meaningfulness until the energy does not decrease anymore.

Figure 8 illustrates the need for contrast inversion invariance in the case of document restoration. Here, the important point is that the CToS is able to retrieve low-contrasted letters even in the presence of "show-through". Since we use a segmentation energy, we do not pretend that it is the perfect solution for document binarization, however since the documents are largely simplified while keeping all the objects of interest, it may serve as a pre-processing for a more specific binarization method.

4.3 Interactive Object Segmentation

In [10], the authors introduced a method for interactive image segmentation using the level set representation of the image. We extend basically the same idea to the CToS. 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 all the other pixels to one of these classes. We then use the Nearest Neighbor classifier where the distance between two points x and y is the minimal total variation along all the all paths from x to y (see Eq. (1)). 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.

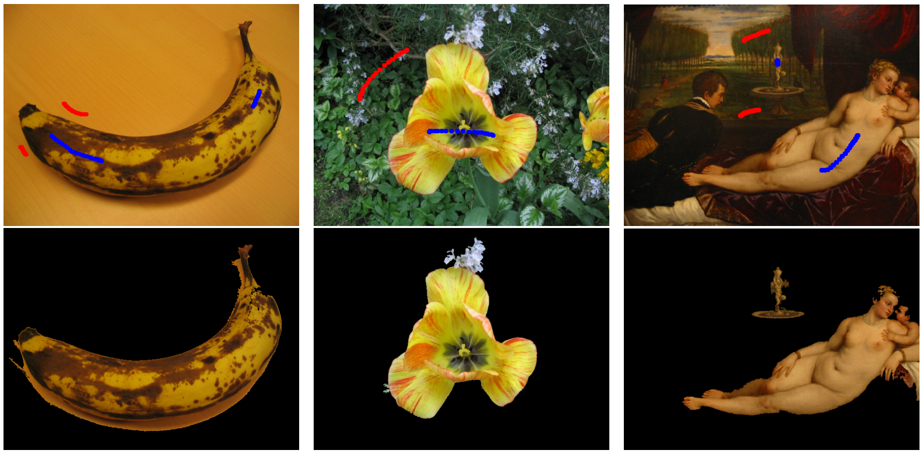


Fig. 9. Interactive segmentation using the CToS. original images with the markers on the top line and the segmentation below.

As a consequence, instead of working at the pixel level, the classification can be done equivalently with the ToS by computing the influence zones of the shapes having a marker pixel using the tree topology. With the CToS, a node may contain pixels of different colors, so we consider that the distance between a shape and its parent is simply the L^2-distance of their average color (in RGB, or better in the La*b* space).

A strong advantage of the method is its ability to recover large regions of interest with very few markers (see Fig. 9) whereas many other methods using statistics require larger markers for a better learning accuracy. We did not show the results using the ToS computed on the luminance only but the same problems (so the same remarks) stand as for the simplification.

5 Conclusion and Perspectives

We have presented a method to extend the ToS on multivariate images. Contrary to standard approaches, our CToS does not rely on any choice of multivariate total ordering but is only based on the inclusion relationship of the shapes and outputs a tree which is marginally both self-dual and contrast change invariant. In this paper, we have tried to highlight why those properties are important for image processing and computer vision tasks. Eventually, we have shown the versatility and the potential of our representation. As perspectives, we will focus on some other kinds of multivariate data such as hyperspectral satellite images or multimodal medical images to validate the contributions of our approach to the processing of such data. Moreover, we also plan to compare the CToS to other hierarchical representations such as hierarchies of partitions (quasi-flat zones hierarchy, binary partition trees…) to further study the pros and cons of our method.

References

1. Aptoula, E., Lefèvre, S.: A comparative study on multivariate mathematical morphology. Pattern Recognition 40(11), 2914–2929 (2007)
2. Cao, F., Musé, P., Sur, F.: Extracting meaningful curves from images. Journal of Mathematical Imaging and Vision 22(2-3), 159–181 (2005)
3. Carlinet, E., Géraud, T.: Getting a morphological tree of shapes for multivariate images: Paths, traps and pitfalls. In: Proc. of ICIP. pp. 615–619. France (2014)
4. Carlinet, E., Géraud, T.: A morphological tree of shapes for color images. In: Proc. of Intl. Conf. on Pattern Recognition (ICPR). pp. 1132–1137. Sweden (Aug 2014)
5. Caselles, V., Coll, B., Morel, J.M.: Geometry and color in natural images. Journal of Mathematic Imaging and Vision 16(2), 89–105 (2002)
6. Caselles, V., Monasse, P.: Geometric Description of Images as Topographic Maps, Lecture Notes in Mathematics, vol. 1984. Springer-Verlag (2009)
7. Caselles, V., Monasse, P.: Grain filters. Journal of Mathematic Imaging and Vision 17(3), 249–270 (Nov 2002)
8. Chevallier, E., Angulo, J.: Image adapted total ordering for mathematical morphology on multivariate image. In: Proceedings of the IEEE International Conference on Image Processing (ICIP). pp. 2943–2947. Paris, France (Oct 2014)
9. Dibos, F., Koepfler, G.: Total variation minimization by the Fast Level Sets Transform. In: IEEE Workshop on Variational and Level Set Methods in Computer Vision. pp. 179–185. IEEE Computer Society (2001)
10. Dubrovina, A., Hershkovitz, R., Kimmel, R.: Image editing using level set trees. In: Proceedings of the IEEE International Conference on Image Processing (ICIP). pp. 4442–4446. Paris, France (Oct 2014)
11. Géraud, T., Carlinet, E., Crozet, S., Najman, L.: A quasi-linear algorithm to compute the tree of shapes of n-D images. In: Proc. of Intl. Symp. on Mathematical Morphology (ISMM). LNCS, vol. 7883, pp. 98–110. Springer (2013)
12. Heijmans, H.J.A.M.: Self-dual morphological operators and filters. Journal of Mathematic Imaging and Vision 6(1), 15–36 (1996)
13. Lezoray, O., Charrier, C., Elmoataz, A., et al.: Rank transformation and manifold learning for multivariate mathematical morphology. In: Proc. of European Signal Processing Conference. vol. 1, pp. 35–39 (2009)
14. Naegel, B., Passat, N.: Towards connected filtering based on component-graphs. In: Proc. of ISMM. LNCS, vol. 7883, pp. 353–364. Springer (2013)
15. Passat, N., Naegel, B.: An extension of component-trees to partial orders. In: Proc. of IEEE Intl. Conf. on Image Processing (ICIP). pp. 3933–3936. IEEE Press (2009)
16. Salembier, P., Serra, J.: Flat zones filtering, connected operators, and filters by reconstruction. IEEE Transactions on Image Processing 4(8), 1153–1160 (1995)
17. Soille, P.: Beyond self-duality in morphological image analysis. Image and Vision Computing 23(2), 249–257 (2005)
18. Tushabe, F., Wilkinson, M.H.F.: Color processing using max-trees: A comparison on image compression. In: Proc. of ICSAI. pp. 1374–1380. IEEE (2012)
19. Velasco-Forero, S., Angulo, J.: Supervised ordering in $\mathcal{R}_p$: Application to morphological processing of hyperspectral images. IEEE Transactions on Image Processing 20(11), 3301–3308 (2011)
20. Velasco-Forero, S., Angulo, J.: Random projection depth for multivariate mathematical morphology. IEEE Journal of Selected Topics in Signal Processing 6(7), 753–763 (2012)
21. Xu, Y.: Tree-based shape spaces: Definition and applications in image processing and computer vision. Ph.D. thesis, Université Paris-Est (Dec 2013)
22. Xu, Y., Géraud, T., Najman, L.: Salient level lines selection using the Mumford-Shah functional. In: Proc. of ICIP. pp. 1227–1231 (2013)