



**HAL**  
open science

# How to modify the tree of shapes of an image: Connected operators without gradient inversion

Julien Mendes Forte, Nicolas Passat, Yukiko Kenmochi

## ► To cite this version:

Julien Mendes Forte, Nicolas Passat, Yukiko Kenmochi. How to modify the tree of shapes of an image: Connected operators without gradient inversion. International Conference on Pattern Recognition (ICPR), 2024, Kolkata, India. hal-04668002v2

**HAL Id: hal-04668002**

**<https://hal.science/hal-04668002v2>**

Submitted on 29 Aug 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# How to Modify the Tree of Shapes of an Image: Connected Operators Without Gradient Inversion<sup>★</sup>

Julien Mendes Forte<sup>1</sup>, Nicolas Passat<sup>2</sup>, and Yukiko Kenmochi<sup>1</sup>

<sup>1</sup> Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, F-14050 Caen, France

<sup>2</sup> Université de Reims Champagne-Ardenne, CReSTIC, 51100 Reims, France

**Abstract.** The tree of shapes is a hierarchical data structure that models a grey-level image via its level lines. It belongs to the family of morphological trees, which allow to design connected operators, i.e. non-linear filters that transform an image without creating new contours. Connected operators act by modifying the image-modeling tree, shifting the values of its nodes. This paradigm is frequently used with the component tree, another popular morphological tree. It is much less considered in the case of the tree of shapes despite its ability to model more finely the image. Indeed, shifting the values of the nodes of a tree of shapes is more complex, compared to other morphological trees. In this article, we investigate how to modify a tree of shapes by shifting the values of its nodes. We explain how to carry out this operation so that the modified / simplified tree remains the tree of shapes of the processed image. We propose algorithmic solutions and methodological schemes to reach that goal. We discuss on their properties and we illustrate their relevance by application examples of induced connected operators. Software implementation available at <https://github.com/jmendesf/ToSConOp>.

**Keywords:** tree of shapes · connected filters · grey-level imaging · hierarchical models · mathematical morphology

## 1 Introduction

Graph-based hierarchical structures are a cornerstone of mathematical morphology. These hierarchical structures—most often trees—allow to model, describe and analyse images. They also provide a way to process images, by allowing the design of connected operators [29], a family of non-linear filters which share the specific property of modifying an image while preserving its contours (i.e. not creating contours that do not exist in the input image). This property derives from the fact that these operators do not act at the scale of the pixels, but at the scale of the connected components / flat zones of the image modeled by a tree [28]. The applications of such operators are many, including mainly filtering and segmentation.

---

<sup>★</sup> Work funded by Région Normandie (thesis grant RIN), Partenariats Hubert Curien (Sakura, 49674RK) and Agence Nationale de la Recherche (ANR-20-CE45-0011, ANR-22-CE45-0034, ANR-23-CE45-0015). ANATOMIX is an EQUIPEX funded by the Investments for the Future program of the Agence Nationale de la Recherche (ANR-11-EQPX-0031).

Two kinds of trees were proposed for modeling images: the component tree [27] and the tree of shapes [17]. Both were initially designed for modeling grey-level images and further generalized to handle multivalued images [14, 7]. The design of connected operators based on such trees relies on a three-step procedure [13]: (1) building the tree from the original image (2) modifying the tree (generally, simplifying it, which can be done by attribute analysis [4] or shaping [32]), and (3) reconstructing the final image from the modified tree. While component tree based connected operators enjoy a wide variety of applications, they are not well-fitted when brighter and darker areas of the images need to be handled simultaneously. In such cases, using both the min- and max-trees in a parallel or iterative fashion may lead to unsatisfactory results. Despite the strong links that exist between the component tree and the tree of shapes [21], and while connected operators based on the tree of shapes have been proposed [33], this three-step procedure was mostly used to develop connected operators based on the component tree.

This fact derives from the distinct modeling proposed by the two kinds of trees. The component tree builds upon the connected components of the binary threshold sets of the image. As consequence, simplifying that tree, i.e. removing a node, provides a valid component tree. It also has a straightforward effect on the image, by lowering the values of the region associated to the node, leading in particular to gradient-sign preserving, anti-extensive filters. By contrast, the tree of shapes builds upon regions bounded by the level lines of the image. As a consequence, simplifying that tree by removing a node (or more generally by modifying its value) provides a tree which may not be a valid tree of shapes. It may also have unexpected effects on the image, in particular by modifying the sign of the gradients of the contours.

To the best of our knowledge, the way to modify the value of the nodes in a tree of shapes without altering its intrinsic properties nor its ability to model the associated image had not been investigated until now [10]. We propose to tackle this issue.

This article is organized as follows. Sect. 2 describes related works on the tree of shapes and connected operators. Sect. 3 provides notions and properties related to the tree of shapes. Sect. 4 describes how an image can be modified by shifting the value of the nodes of its tree of shapes, and discusses on the related issues. Sect. 5 provides an algorithm for shifting the value of one node of a tree of shapes whereas guaranteeing that it remains the tree of shapes of the modified image. Sect. 6 proposes a generic scheme that extends the one-node shifting algorithm as a many-node shifting. Sect. 7 exemplifies this scheme with induced connected operators. Sect. 8 concludes the article.

## 2 Related works

The trees developed in mathematical morphology are subdivided into two families. The first gathers the total partition trees (e.g. the binary partition tree [26] and the watershed tree [18]) that decompose the image into incrementally refining partitions. The second gathers the partial partition trees that model images with respect to their spatial-spectral structure. The tree of shapes [17] belongs to this second family. It models a grey-level image by considering it as a topographic map. It encodes the nested relation between

the isocontours of the image. It is the grey-level generalization of the adjacency tree [24] that models the topological structure of a binary image.

Modeling an image by a tree of shapes requires to fulfill specific topological constraints. Indeed, the support of the image has to satisfy the Jordan-Brouwer property. This is guaranteed for certain topological frameworks, e.g. for the well-composed images [15]. Digitally well-composed interpolations were investigated for the definition of trees of shapes in arbitrary dimensions [3].

Variants of the tree of shapes were proposed. One can cite the multivariate tree of shapes [7] that handles non-grey-level images, or the topological monotonic tree [30] and the topological tree of shapes [20], that focus on the topological structure of the tree of shapes. The links that exist between the tree of shapes and the component tree [27] (the other archetype of the partial partition trees) were also investigated. These links are known since their introduction, and were mainly characterized by the hole-filling procedure between the nodes of the component trees and those of the tree of shapes. Recently, the homeomorphic links between both trees were explicitated and formalized [21].

Efforts were geared towards efficiently building the tree of shapes. A method based on immersing the image domain in the Khalimsky grid guarantees a worst-case quasilinear complexity [12]. Alternative approaches, including a root-to-leaves paradigm [16], have also been presented, and parallel strategies were investigated [11]. The literature on the construction of the component tree and the tree of shapes proposes some algorithms that rely on the first to build the second [9, 5] or vice versa [31].

The rich information modeled by the tree of shapes allowed the development of various image processing and analysis approaches: filtering [8], segmentation [1], simplification [2] or object recognition [19]. The image processing methods based on the tree of shapes belong to the family of the connected operators [29]. Such operators model an image via a morphological tree (tree of shapes, component tree), modify that tree, and reconstruct the image accordingly [13]. The modification of the tree is most often a simplification that discards the nodes of the tree that do not satisfy a given criterion, based on descriptive attributes [4].

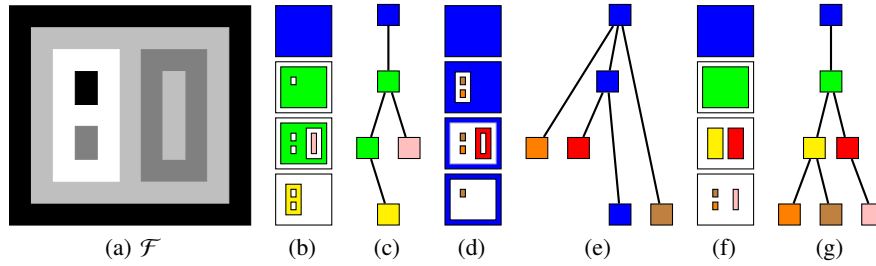
Connected operators based on this paradigm mainly build upon component trees, generally leading to anti-extensive filters. This sheds light on the spectral anisotropy of these component tree-based operators. By contrast, the simplification (or more generally, the modification) of a tree of shapes generally belongs to the class of self-dual connected operators, leading to an isotropic behaviour, relevant for many image processing applications. Such tree-of-shapes-based connected operators are especially interesting as they can take into account both minimal (“dark”) and maximal (“light”) regions of the image. However, modifying / simplifying a tree of shapes in a convenient fashion is an operation that is more complex than for a component tree (see Sect. 4).

### 3 Tree of shapes

#### 3.1 Basic definitions

An image is defined as a function  $\mathcal{F} : \mathbb{U} \rightarrow \mathbb{V}$  (Fig. 1(a)). We assume that (1)  $\mathbb{U}$  is endowed with a topological structure compliant with the Jordan-Brouwer property and





**Fig. 1.** An image  $\mathcal{F}$  (a), its upper- (b) and lower-threshold sets (d) (coloured regions). The two component trees of  $\mathcal{F}$ : the max-tree (c) that derives from (b) and the min-tree (e) that derives from (d). Both model the inclusion between connected components. The tree of shapes (g) that derives from the hole filling of the connected components of (b,d). The connected components of (b,d) of same color lead to the node of this color in (f,g).

(2)  $\mathbb{V}$  is endowed with a total order relation  $\leq$ . For the sake of readability, we set  $\mathbb{U} = \mathbb{Z}^d$  ( $d \geq 2$ ) endowed with the digital topology framework [25]. We also set  $\mathbb{V} = \mathbb{Z}$  so that  $\mathcal{F}(\mathbb{U}) = \llbracket \perp, \top \rrbracket \subset \mathbb{V}$  and a finite number of points  $\mathbf{x} \in \mathbb{U}$  satisfy  $\mathcal{F}(\mathbf{x}) > \perp$ . These hypotheses are generally satisfied by digital images.

Let  $v \in \mathbb{V}$ . The upper- and lower-threshold sets of  $\mathcal{F}$  at value  $v$  (Fig. 1(b,d)) are the subsets of  $\mathbb{U}$  defined as

$$\Lambda_v^\circ(\mathcal{F}) = \{\mathbf{x} \in \mathbb{U} \mid v \leq \mathcal{F}(\mathbf{x})\} \quad \text{and} \quad \Lambda_v^*(\mathcal{F}) = \{\mathbf{x} \in \mathbb{U} \mid v > \mathcal{F}(\mathbf{x})\} \quad (1)$$

Let  $A \subseteq \mathbb{U}$ . We note  $\Pi[A] \subseteq 2^{\mathbb{U}}$  the set of the connected components of  $A$  (we set  $\Pi[A] = \emptyset$  if  $A = \emptyset$ ). We note

$$C^\circ = \bigcup_{v \in \mathbb{V}} \Pi[\Lambda_v^\circ(\mathcal{F})] \quad \text{and} \quad C^* = \bigcup_{v \in \mathbb{V}} \Pi[\Lambda_v^*(\mathcal{F})] \quad (2)$$

the set of the connected components of  $\mathcal{F}$  induced by its upper- and lower-threshold sets, respectively (Fig. 1(b,d)).

Let  $Z \in \Pi[A]$ . We note  $Z^\tau \supseteq Z$  the connected set obtained by filling the cavities of  $Z$  (Fig. 1(f)). We set  $C = C^\circ \cup C^*$  and

$$\Theta = \{Z^\tau \mid Z \in C\} \quad (3)$$

We consider the partial order relation  $\subseteq$  on  $\Theta$  and we note  $\triangleleft$  its reflexive-transitive reduction. The tree of shapes (Fig. 1(g)) is then defined as follows.

**Definition 1 (Tree of shapes [17])** *The tree of shapes of  $\mathcal{F}$  is the tree  $\mathfrak{T} = (\Theta, \triangleleft)$ .*

The elements of  $\Theta$  (resp.  $\triangleleft$ ) are called the *nodes* (resp. the *edges*) of  $\mathfrak{T}$ .

### 3.2 Proper part, altitude, image reconstruction

Each node  $X \in \Theta$  is characterized spatially (with respect to  $\mathbb{U}$ ) and spectrally (with respect to  $\mathbb{V}$ ). This leads to the two notions of *proper part* and the *altitude* of a node.

**Definition 2 (Proper part of a node)** Let  $X \in \Theta$ . The proper part of  $X$  in the tree of shapes  $\mathfrak{T} = (\Theta, \triangleleft)$  is defined by

$$\rho(X) = X \setminus \bigcup_{Y \triangleleft X} Y \quad (4)$$

We note  $\rho : \Theta \rightarrow 2^{\mathbb{U}}$  the induced function.

**Remark 3** The set  $\{\rho(X) \mid X \in \Theta\}$  is a partition of  $\mathbb{U}$ .

**Definition 4 (Altitude of a node)** Let  $X \in \Theta$ . The altitude  $Alt(X)$  of  $X$  in the tree of shapes  $\mathfrak{T} = (\Theta, \triangleleft)$  is defined by

$$\forall \mathbf{x} \in \rho(X), Alt(X) = \mathcal{F}(\mathbf{x}) \quad (5)$$

The altitude  $Alt(X)$  of  $X$  is the unique value of  $\mathbb{V}$  that  $\mathcal{F}$  assigns to the points of the proper part  $\rho(X)$  of  $X$ . We note  $Alt : \Theta \rightarrow \mathbb{V}$  the induced function.

From now on, we consider that the tree of shapes  $\mathfrak{T}$  is implicitly endowed with its induced proper part and altitude functions, i.e. we consider  $\mathfrak{T}$  as  $(\mathfrak{T}, \rho, Alt)$ . From  $(\mathfrak{T}, \rho, Alt)$ , it is possible to reconstruct the image in a lossless way.

**Property 5** Given the tree of shapes  $(\mathfrak{T} = (\Theta, \triangleleft), \rho, Alt)$ , the image  $\mathcal{F} : \mathbb{U} \rightarrow \mathbb{V}$  can be recovered by setting, for any  $\mathbf{x} \in \mathbb{U}$

$$\mathcal{F}(\mathbf{x}) = \bigvee_{X \in \Theta}^{\leq} \mathbf{1}_{(\rho(X), Alt(X))}(\mathbf{x}) \quad (6)$$

where  $\bigvee^{\leq}$  is the supremum for  $\leq$ , and  $\mathbf{1}_{(Y,u)} : \mathbb{U} \rightarrow \mathbb{V}$  is the cylinder function defined by  $\mathbf{1}_{(Y,u)}(\mathbf{x}) = u$  if  $\mathbf{x} \in Y$  and  $\perp$  otherwise.

In other words, the image  $\mathcal{F}$  can be reconstructed by assigning to each points  $\mathbf{x} \in \mathbb{U}$  the altitude value  $Alt(X)$  of the node  $X \in \Theta$  that contains  $\mathbf{x}$  in its proper part  $\rho(X)$ . This reconstruction formula constitutes the cornerstone for designing connected operators based on the tree of shapes of an image, as discussed in the next section.

## 4 How to modify an image via its tree of shapes?

Before discussing on the connected operators based on morphological trees (Sect. 4.2), we recall properties of the tree of shapes related to its invariance to contrast transformations (Sect. 4.1).

### 4.1 Contrast invariance of the tree of shapes

Let  $\gamma : \mathbb{V} \rightarrow \mathbb{V}$  be a transformation of the space of values. We say that  $\gamma$  is a contrast transformation if it is strictly increasing, i.e. if for all  $u, v \in \mathbb{V}$  we have  $(u < v) \Rightarrow (\gamma(u) < \gamma(v))$ .

**Property 6 (Contrast invariance of the tree of shapes)** *Let  $\mathcal{F} : \mathbb{U} \rightarrow \mathbb{V}$  be an image. Let  $\gamma : \mathbb{V} \rightarrow \mathbb{V}$ . Let  $(\mathfrak{T}, \rho, Alt)$  be the tree of shapes of  $\mathcal{F}$ . If  $\gamma$  is a contrast transformation, then  $(\mathfrak{T}, \rho, \gamma \circ Alt)$  is the tree of shapes of  $\gamma \circ \mathcal{F}$ .*

In other words, a contrast transformation  $\gamma$  alters neither the structural ( $\mathfrak{T}$ ) nor the spatial ( $\rho$ ) part of the tree of shapes of an image, whereas the spectral part ( $\gamma \circ Alt$ ) directly follows the contrast transformation.

A contrast transformation  $\gamma$  acts on the space of values  $\mathbb{V}$ . It induces a contrast transformation  $\Gamma : \mathbb{V}^{\mathbb{U}} \rightarrow \mathbb{V}^{\mathbb{U}}$  that acts on the space of images. It is defined, for any image  $\mathcal{F} : \mathbb{U} \rightarrow \mathbb{V}$ , by  $\Gamma(\mathcal{F}) = \gamma \circ \mathcal{F}$ . The function  $\Gamma$  exhibits an important property: it is gradient-sign preserving. Let  $\mathbf{x}, \mathbf{y} \in \mathbb{U}$  such that  $\mathbf{x}$  and  $\mathbf{y}$  are neighbour with respect to the topological structure of  $\mathbb{U}$ . Let  $\mathcal{F} : \mathbb{U} \rightarrow \mathbb{V}$  be an image. The gradient sign-preserving property of  $\Gamma$  means that

$$(\mathcal{F}(\mathbf{x}) \leq \mathcal{F}(\mathbf{y})) \implies (\Gamma(\mathcal{F}(\mathbf{x})) \leq \Gamma(\mathcal{F}(\mathbf{y}))) \quad (7)$$

As a corollary of Eq. (7), we have

$$(\mathcal{F}(\mathbf{x}) = \mathcal{F}(\mathbf{y})) \implies (\Gamma(\mathcal{F}(\mathbf{x})) = \Gamma(\mathcal{F}(\mathbf{y}))) \quad (8)$$

From Eq. (8),  $\Gamma$  is a connected operator, i.e. it does not create new contours in the transformed image.

From Props. 5–6, the application of  $\Gamma$  on the image  $\mathcal{F}$  could be carried out by (1) building the tree of shapes  $(\mathfrak{T}, \rho, Alt)$  of  $\mathcal{F}$ ; (2) modifying  $Alt$  into  $\gamma \circ Alt$ ; (3) reconstructing the image  $\Gamma(\mathcal{F})$  modeled by  $(\mathfrak{T}, \rho, \gamma \circ Alt)$  (see Eq. (6)).

For a trivial transformation such as  $\Gamma$ , acting directly on  $\mathcal{F}$  would be sufficient. Nonetheless, the three-step procedure inspired from [13] and discussed in Sect. 4.2 opens the way to the design of a wider range of gradient-sign preserving connected operators based on the modification of the values of the nodes of the tree of shapes, leading to *local* modifications of the contrast of the modeled images.

## 4.2 Connected filtering framework: issues and purpose

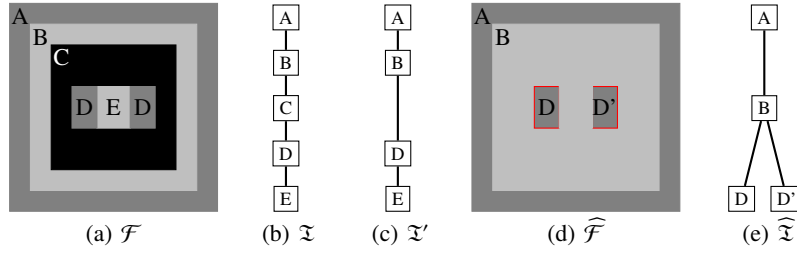
As stated above, a connected operator can be designed from the simplification of a tree. The process consists in:

$$\begin{array}{ccc}
 \mathcal{F} & \xrightarrow{(i)} & \mathfrak{T} \\
 \downarrow \Gamma & & \downarrow (ii) \\
 \widehat{\mathcal{F}} & \xleftarrow{(iii)} & \widehat{\mathfrak{T}}
 \end{array} \quad (9)$$

(i) building the tree  $\mathfrak{T}$  that models the image  $\mathcal{F}$ ;  
(ii) simplifying  $\mathfrak{T}$  into a new tree  $\widehat{\mathfrak{T}}$ ; and  
(iii) reconstructing the new image  $\widehat{\mathcal{F}}$  from  $\widehat{\mathfrak{T}}$ .

It was pioneered in [13], where it was designed for component trees [27] and by considering that the simplification of Step (ii) consists of discarding nodes according to attributes [4].

This framework may be considered also with a tree of shapes instead of a component tree. Step (i) relies on the construction of the tree, which can be done either for the component tree [6] or the tree of shapes [12]. Step (iii) relies on the reconstruction formula of Eq. (6), which is similar for both the component tree and the tree of shapes.



**Fig. 2.** An image  $\mathcal{F}$  (a) and its tree of shapes  $\mathfrak{T}$  (b). The removal of C (equivalent to setting its altitude to that of B) leads to the tree  $\mathfrak{T}'$  (c). The image obtained from  $\mathfrak{T}'$  (d) and its tree of shapes  $\widetilde{\mathfrak{T}}$  (e). (1) The tree  $\mathfrak{T}'$  (c) differs from  $\widetilde{\mathfrak{T}}$  (e). (2) The sign of the gradients has been modified between  $\mathcal{F}$  and  $\widehat{\mathcal{F}}$  for some contours (d, in red). (Figure adapted from [17].)

Regarding Step (ii), in [13] and most of the subsequent contributions, the idea was to preserve some nodes of the component tree whereas discarding others. In particular, when a node  $X$  was preserved,  $\rho(X)$  and  $Alt(X)$  were unaltered. By contrast, when a node  $X$  was discarded,  $\rho(X)$  was merged with the proper part  $\rho(P)$  of its parent node  $P$  such that  $X < P$ . This discarding / merging can be seen as a side effect of the modification of the altitude  $Alt(X)$  so that it becomes equal to  $Alt(P)$ . Step (ii) could then be generalized as “turn  $\mathfrak{T}$  into a new tree  $\widetilde{\mathfrak{T}}$  by modifying the altitudes of its nodes”.

Defining a new altitude function  $\widehat{Alt} : \Theta \rightarrow \mathbb{V}$  does not modify the structure of the tree  $\mathfrak{T}$ . Besides, the reconstruction formula of Eq. (6) remains valid, making Step (iii) tractable. The induced operator  $\Gamma$  applied on  $\mathcal{F}$  is a connected operator, since it satisfies Eq. (8). Nonetheless, two problems may occur (Fig. 2). First, the tree  $\mathfrak{T}'$  associated to the new altitude application  $\widehat{Alt}$ , and to the induced image  $\widehat{\mathcal{F}}$  may not be the tree of shapes  $\widetilde{\mathfrak{T}}$  of  $\widehat{\mathcal{F}}$ . More generally, it may not be a tree of shapes (Fig. 2(c,e)). Second, the operator  $\Gamma$ , although being a connected operator, may not satisfy Eq. (7), i.e. it may not be gradient sign-preserving (Fig. 2(a,d)).

We propose hereafter a framework that allows to modify the altitude of the nodes of a tree of shapes—with controlled side effects on its structure—so that:

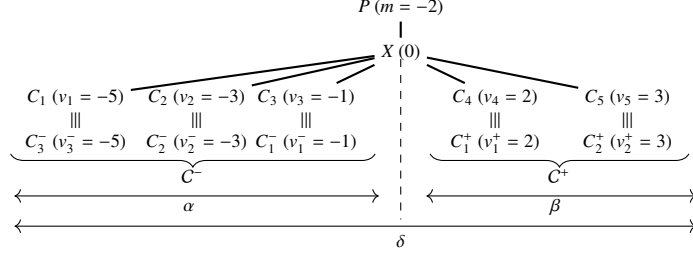
- ( $P_1$ ) the modified tree remains the tree of shapes of the associated modified image;
- ( $P_2$ ) the sign of the gradients of the modified image is preserved with respect to the initial image.

## 5 Shifting one node of the tree of shapes

We now explain how to modify the altitude of (i.e. *shift*) a node of a tree of shapes while satisfying ( $P_1$ ) and ( $P_2$ ).

### 5.1 Notations

Let  $\mathcal{F} : \mathbb{U} \rightarrow \mathbb{V}$  be an image. Let  $\mathfrak{T} = (\Theta, <)$  be its tree of shapes. Let  $X \in \Theta$  be a node of  $\mathfrak{T}$ . From Prop. 6, we assume without loss of generality that  $Alt(X) = 0$ . (See Fig. 3.)



**Fig. 3.** Graphical example of notations of Sect. 5.1. The altitudes of nodes are given into brackets. In this figure, we have  $m^+ = v_4$  and  $m^- = v_3$ . They correspond to the closest (higher and lower) altitudes with respect to  $X$  within the set of its children. We have  $M^+ = \{C_4\}$  and  $M^- = \{C_3\}$ , which are the sets of children nodes associated to the respective values  $m^+$  and  $m^-$ . Finally, we have  $\mu^+ = v_4$  and  $\mu^- = v_3$  which correspond to the closest (higher and lower) altitudes with respect to  $X$  within the whole set of its neighbouring nodes (parent and children).

**Definition 7 (Parent set)** The parent set of  $X$  is  $\mathcal{P}(X) = \{P \in \Theta \mid X \triangleleft P\}$ .

If  $\mathcal{P}(X) = \{P\}$ , then  $P$  is the parent node of  $X$ , i.e.  $X \triangleleft P$ . In that case, by abuse of notation, we write  $\mathcal{P}(X) = P$  and we note  $m = \text{Alt}(P)$ . If  $\mathcal{P}(X) = \emptyset$ , then  $X$  is the root of  $\mathfrak{T}$  and we set  $m = 0$ .

**Definition 8 (Children set)** The children set of  $X$  is  $C(X) = \{C_i \in \Theta \mid C_i \triangleleft X\}_{i=1}^\delta$  (with  $\delta \geq 0$ ).

For any  $i \in \llbracket 1, \delta \rrbracket$ , we note  $v_i$  the altitude value of the node  $C_i$ , i.e.  $v_i = \text{Alt}(C_i)$ . We have  $v_i \neq 0$ , i.e. either  $v_i > 0$  or  $v_i < 0$ .

We assume that  $\{C_i\}_{i=1}^\delta$  is sorted with respect to the values  $v_i$ , i.e. for all  $j, k \in \llbracket 1, \delta \rrbracket$ , we have

$$(j < k) \implies (v_j \leq v_k) \quad (10)$$

We set  $\alpha = |\{C_i \mid v_i < 0\}|$  and  $\beta = |\{C_i \mid v_i > 0\}|$  (with  $\delta = \alpha + \beta$ ). In particular, we have  $\{C_i \mid v_i < 0\} = \{C_i\}_{i=1}^\alpha$  and  $\{C_i \mid v_i > 0\} = \{C_i\}_{i=\alpha+1}^{\alpha+\beta}$ . For each  $i \in \llbracket 1, \alpha \rrbracket$ , we set  $C_i^- = C_{\alpha-i+1}$  and we note  $v_i^- = v_{\alpha-i+1}$ . For each  $i \in \llbracket 1, \beta \rrbracket$ , we set  $C_i^+ = C_{\alpha+i}$  and we note  $v_i^+ = v_{\alpha+i}$ . We set  $C^- = \{C_i^-\}_{i=1}^\alpha$  and  $C^+ = \{C_i^+\}_{i=1}^\beta$ . We set

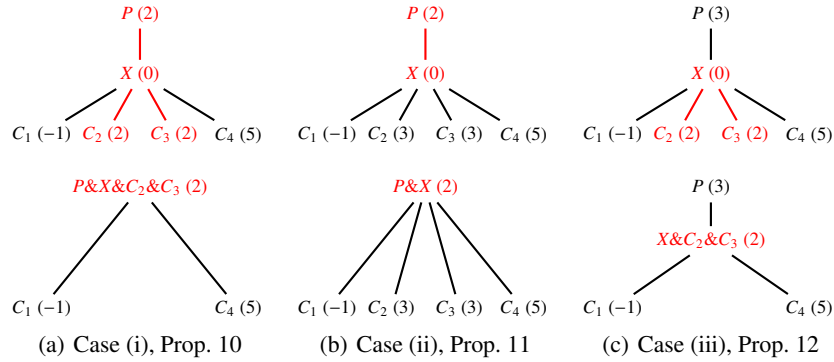
$$m^+ = \min\{v_i^+\}_{i=1}^\beta = \begin{cases} v_1^+ & \text{if } \beta > 0 \\ +\infty & \text{if } \beta = 0 \end{cases} \quad \text{and} \quad m^- = \max\{v_i^-\}_{i=1}^\alpha = \begin{cases} v_1^- & \text{if } \alpha > 0 \\ -\infty & \text{if } \alpha = 0 \end{cases} \quad (11)$$

and

$$\mu^+ = \begin{cases} \min\{m^+, m\} & \text{if } m > 0 \\ m^+ & \text{if } m \leq 0 \end{cases} \quad \text{and} \quad \mu^- = \begin{cases} \max\{m^-, m\} & \text{if } m < 0 \\ m^- & \text{if } m \geq 0 \end{cases} \quad (12)$$

Finally, we set

$$\mathcal{M}^+ = \{C_i^+ \in C^+ \mid v_i^+ = \mu^+\} \quad \text{and} \quad \mathcal{M}^- = \{C_i^- \in C^- \mid v_i^- = \mu^-\} \quad (13)$$



**Fig. 4.** The three cases of short range shifting. First row: initial configuration. Second row: final configuration. (a–c) We have  $\mu^- = -1, \mu^+ = 2$  and the targeted altitude is  $v = 2$ . The part of the tree impacted by the shifting is depicted in red.

## 5.2 Short range shifting

Let us first suppose that we want to shift a node  $X \in \Theta$  with no impact beyond its immediate vicinity. The following property states that if this shifting is small enough, then the tree of shapes remains unchanged.

**Property 9** *If the new value  $v$  of  $\text{Alt}(X)$  satisfies  $\mu^- < v < \mu^+$ , then neither  $\mathfrak{T}$  nor  $\rho$  are modified, and  $\text{Alt}$  is modified only for  $X$ .*

Now, let us suppose that we want to shift  $X$  slightly further. More precisely, we want to set the new altitude  $v$  of  $X$  so that  $v = \mu^\star$  (where  $\star$  is either  $+$  or  $-$ ). Three (mutually exclusive) distinct cases can occur (Fig. 4):

- (i)  $\mu^\star = m = m^\star$  (see Prop. 10);
- (ii)  $\mu^\star = m \neq m^\star$  (see Prop. 11);
- (iii)  $\mu^\star = m^\star \neq m$  (see Prop. 12).

In each case, it is necessary to discard nodes, as the new altitude  $v$  of  $X$  is either that of its parent (ii), of some of its children (iii), or both (i). The node discarding procedure is defined in Func. Discard. For a node  $X \in \Theta$  to be discarded, it proceeds as follows:  $\rho(X)$  is added to the proper part of the parent  $P$  of  $X$  (Line 1);  $X$  is removed from  $\Theta$  (Line 2);  $(X, P)$  is removed from  $\triangleleft$  (Line 3); for each children  $C$  of  $X$ ,  $(C, X)$  is removed from  $\triangleleft$  and replaced by  $(C, P)$  (Lines 4–6). From now on, “Discarding a node  $X$ ” will mean that we apply Func. Discard for  $X$ .

Alg. 1 provides the general short range shifting process that corresponds to Cases (i–iii) and Props. 10–12.

**Property 10** *If the new value  $v$  of  $X$  is  $\mu^\star = m = m^\star$ , then the tree of shapes of the image  $\mathcal{F}$  is modified as follows: (1) each  $C \in \mathcal{M}^\star$  is discarded; (2)  $X$  is discarded.*

In that case, the targeted altitude of  $X$  reaches both the altitude of some of its children and of its parent. We discard these children (Alg. 1, Lines 4–5) and then  $X$  (Alg. 1, Line 6).

---

**Function Discard**

---

**Input:**  $\Theta, \triangleleft, \rho, X$   
**Output:**  $\Theta, \triangleleft, \rho$

- 1  $\rho(P) := \rho(P) \cup \rho(X)$
- 2  $\Theta \rightarrow X$
- 3  $\triangleleft \rightarrow (X, P)$
- 4 **foreach**  $C \triangleleft X$  **do**
- 5  $\triangleleft \rightarrow (C, X)$
- 6  $\triangleleft \leftarrow (C, P)$

---



---

**Algorithm 1: Short range shifting of a node**

---

**Input:**  $\Theta, \triangleleft, \rho, Alt, X, v \in \llbracket \mu^-, \mu^+ \rrbracket$   
**Output:**  $\Theta, \triangleleft, \rho, Alt$

- 1 **if**  $\mu^- < v < \mu^+$  **then**  $Alt(X) := v$  (Prop. 9)
- 2 **else**
- 3 **if**  $v = m^*$  **then**
- 4 **foreach**  $S \in \mathcal{M}^*$  **do**  $Discard(\Theta, \triangleleft, \rho, S)$  (Cases (i), (iii), Props. 10, 12)
- 5  $Alt(X) := v$
- 6 **if**  $v = m$  **then**  $Discard(\Theta, \triangleleft, \rho, X)$  (Cases (i), (ii), Props. 10, 11)

---



---

**Algorithm 2: Long range shifting**

---

**Input:**  $\Theta, \triangleleft, \rho, Alt, X, v$   
**Output:**  $\Theta, \triangleleft, Alt, \rho$

- 1 **while**  $Alt(X) \neq v$  **do**
- 2  $P := \mathcal{P}(X)$
- 3 **if**  $v > \mu^+$  **then**  $b := \mu^+$
- 4 **else if**  $v < \mu^-$  **then**  $b := \mu^-$
- 5 **else**  $b := v$
- 6 Apply **Algorithm 1** ( $\Theta, \triangleleft, \rho, Alt, X, b$ )
- 7 **if**  $X \notin \Theta$  **then**  $X := P$

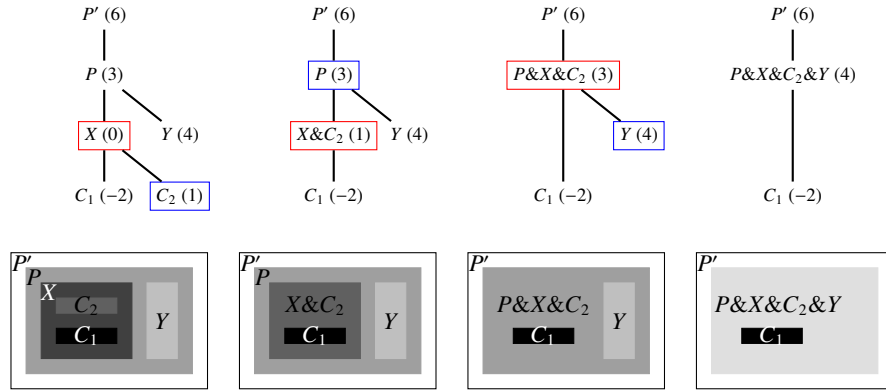
---

**Property 11** *If the new value  $v$  of  $X$  is  $\mu^* = m \neq m^*$ , then the tree of shapes of the image  $\mathcal{F}$  is modified as follows:  $X$  is discarded.*

In that case, the targeted altitude of  $X$  reaches that of its parent only. We discard  $X$  (Alg. 1, Line 6).

**Property 12** *If the new value  $v$  of  $X$  is  $\mu^* = m^* \neq m$ , then the tree of shapes of the image  $\mathcal{F}$  is modified as follows: each  $C \in \mathcal{M}^*$  is discarded.*

In that case, the targeted altitude of  $X$  reaches that of some of its children only. We discard these children (Alg. 1, Lines 4–5).



**Fig. 5.** The successive steps of long-range shifting for a node  $X$  (first row) and the side effect on the image (second row). The target altitude is  $v = 4$ . The red nodes are shifted, while the blue nodes are involved in the intermediate short-range shiftings.

### 5.3 Long range shifting of a node

Let us now suppose that we want to shift a node  $X \in \Theta$  by modifying its altitude  $Alt(X)$  to a value  $v \notin \llbracket \mu^-, \mu^+ \rrbracket$ . We have either  $v > \mu^+$  or  $v < \mu^-$ . Let us suppose that  $v > \mu^+$  (the same reasoning holds for  $v < \mu^-$ ). If we apply Alg. 1 to the node  $X$  to modify its altitude to the value  $v = \mu^+$ , then the node  $X$  is merged with some of its neighbouring nodes (parent and/or children), leading to a new node  $\widehat{X}$  that differs from  $X$  but such that  $X \subset \widehat{X}$ . Thus, further modifying  $X$  boils down to modifying  $\widehat{X}$ . This node  $\widehat{X}$  is associated to its own interval  $\llbracket \hat{\mu}^-, \hat{\mu}^+ \rrbracket$  with  $\mu^+ < \hat{\mu}^+$ . If  $v \leq \hat{\mu}^+$ , then the process ends by applying Alg. 1 (Line 1, Prop. 9) with the value  $v$ . If  $v > \hat{\mu}^+$ , then the process continues one step further by applying Alg. 1 (Lines 3–6, Props. 10–12) with the value  $\hat{\mu}^+$ , and so on. This iterative process is described in Alg. 2 (see Fig. 5).

## 6 Shifting many nodes of the tree of shapes

In Sect. 5, we explained how to shift a single node of a tree of shapes. We now focus on the issue of shifting many nodes.

### 6.1 Criterion and policy

A criterion is a Boolean function  $Crit : \Theta \rightarrow \{True, False\}$  defined on the nodes of the tree of shapes (or, more generally, on any subset of  $\mathbb{U}$ ). Its definition guides the selection of the nodes to be shifted. Such criteria, already considered in [13], often rely on attributes [4] that describe specific (spatial, spectral...) properties of the nodes. Examples of criteria are given in Sect. 7 for the illustrative applications.

The policy of a tree-of-shapes modification framework is defined as a function  $Pol : \Theta \rightarrow \mathbb{V}$ . Given a criterion  $Crit$ , for any node  $X \in \Theta$  such that  $Crit(X) = True$ , the



---

**Algorithm 3:** Generic tree of shapes modification

---

**Input:**  $\Theta, \triangleleft, Alt, \rho$   
**Parameters:**  $Crit : \Theta \rightarrow \{True, False\}, Pol : \Theta \rightarrow \mathbb{V}$   
**Output:**  $\Theta, \triangleleft, Alt, \rho$

- 1 **foreach**  $X \in \Theta$  **such that**  $Crit(X) = True$  **do**
- 2      $v := Pol(X)$
- 3     Apply **Algorithm 2** ( $\Theta, \triangleleft, \rho, Alt, X, v$ )

---

function  $Pol$  is used to assign a new value  $v$  to  $Alt(X)$ . Many tree modification methods proposed in the literature (especially for the component trees) usually aimed at merging a node  $X$  with its parent node  $P$ . In that case, the function  $Pol$  was simply defined so that  $Pol(X) = Alt(P)$ . Here, the considered functions  $Pol$  are more versatile. Examples of policies are given in Sect. 7 for the illustrative applications.

## 6.2 General approach for shifting a set of nodes

Algorithm 3 defines a generic tree of shapes modification framework. The application of  $Crit$  on the nodes of  $\Theta$  defines a set  $\mathcal{N} = \{X \in \Theta \mid Crit(X) = True\}$  (Line 1). Each node  $X \in \mathcal{N}$  is assigned a new altitude  $v \in \mathbb{V}$  using the function  $Pol$  (Line 2) and Alg. 2 is then called in order to effectively shift  $X$  with respect to  $v$  (Line 3).

To keep this framework as generic as possible, we did not propose any sorting of  $\mathcal{N}$  nor tree-traversal strategy. Of course, such choices have a fundamental effect on the behaviour of the process, since the shifting of a node  $X \in \Theta$ —and its possible side effects on the remainder of the tree—may (1) modify the part of  $\mathcal{N}$  not yet processed (e.g. by adding or removing nodes  $Y$  due to modifications of  $Crit(Y)$ ) and (2) impact the way to shift these nodes (e.g. if  $Pol(Y)$  has been impacted by the processing of  $X$ ). Based on these considerations, the way to order  $\mathcal{N}$ , the choice to update (or not) the values of  $Crit$  and  $Pol$  during the process and the way of processing the nodes (e.g. sequentially or simultaneously) are important hyperparameters that have to be set depending on the application and its purpose.

## 6.3 Preservation

Algorithms 3, 2, 1 and Func. Discard incrementally build upon each other. By definition, Func. Discard preserves  $(P_1)$  and  $(P_2)$ . From Props. 9–12, Alg. 1 also preserves  $(P_1)$  and  $(P_2)$ . By construction this is also the case for Alg. 2 and then for Alg. 3.

Moreover, depending on the used function  $Pol$ , additional properties may be satisfied. For instance, by designing  $Pol$  so that for any  $X \in \Theta$ ,  $Pol(X) < Alt(X)$  (resp.  $Pol(X) > Alt(X)$ ), one may build extensive (resp. anti-extensive) connected operators. The versatility of the proposed framework allows to tackle various image processing issues, as illustrated in the next section.

## 6.4 Computational aspects

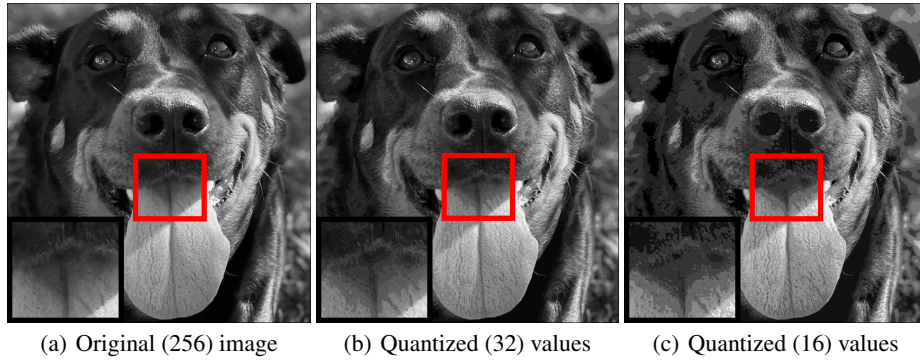
If we assume that the number of children is  $O(1)$  for the nodes of a tree of shapes (which is generally the case), then the time cost of *Funct. Discard* is  $O(1)$ , the time cost of Alg. 1 is  $O(1)$  and the time cost of Alg. 2 is  $O(k)$  where  $k$  is the number of intermediate altitudes the current node has to go through. At each application of Alg. 2, the number of nodes of the tree is reduced by at least  $k$ . It follows that, except the extra-cost induced by the computation of *Crit* and *Pol* in Alg. 3, its overall time cost is  $O(|\Theta|)$ . The complexity of this generic approach described in Alg. 3 then depends on the strategies adopted by the user for defining and updating *Crit* and *Pol*. For well-chosen strategies (e.g. construction of attributes based on separable properties, no updating of the modified nodes. . .), the overall process may be carried out in linear time. Since the construction of the tree of shapes is itself a quasi-linear time process, this may allow the development of efficient strategies, in particular for handling large-scale images. Considering that the simplification process described in Alg. 2 acts in only a part of the tree, composed of the subtree rooted at the processed node plus its upper branch, it is also possible to rely on parallel approaches, that may be deployed on distributed architectures.

## 7 Application examples

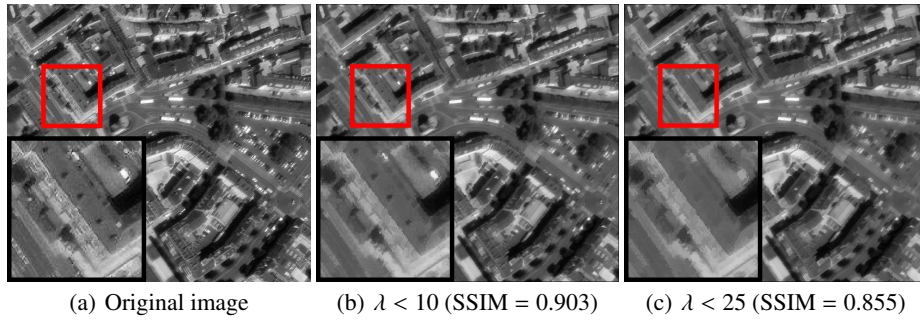
We illustrate the relevance of the proposed approach for tree of shapes modification by providing three examples of induced connected operators acting on grey-level images. In these experiments, the trees of shapes were natively computed from Hgra [22], and subsequently processed by our code (<https://github.com/jmendesf/ToSConOp>).

- The first one performs image quantization (Fig. 6). It reduces the dynamics of an image from  $2^p$  to  $2^k$  grey-levels, ( $0 \leq k \leq p$ ). Here, *Crit* always holds *True*. The quantized grey-levels are regularly sampled over  $\llbracket 0, 2^p - 1 \rrbracket$  and *Pol* is defined so that the altitude of each node is shifted to the closest quantized value. This first application is a toy example, since pixel-wise quantization also satisfies  $(P_1)$  and  $(P_2)$ . It mainly aims to show the generality of our approach.
- The second one performs area opening (Fig. 7). It removes the nodes of the tree with smallest proper parts and, equivalently, the smallest details in the image. Here,  $Crit(X) = True$  iff  $\rho(X) < \lambda$  with  $\lambda \in \mathbb{N}$ . *Pol* is defined so that the altitude of  $X$  be shifted to  $m$ . The nodes  $X$  are processed by increasing size of their proper part.
- The third one performs mean filtering at the scale of the nodes of the tree (Fig. 8). It progressively smoothes the altitudes / grey levels between neighbouring nodes / flat zones. Here, *Crit* always holds *True*. *Pol* is defined so that  $X$  is shifted to an altitude defined as the mean value of the altitude of its neighbouring nodes. The nodes  $X$  are processed from the root to the leaves.

By definition, in each one of these applications, the designed connected operator modifies / simplifies the tree of shapes and the induced images while satisfying  $(P_1)$  and  $(P_2)$ , thus leading to the preservation of the structure of the image.



**Fig. 6.** (a) A natural image (256 grey levels). (b,c) The result of its quantization to 32 and 16 grey levels, respectively.

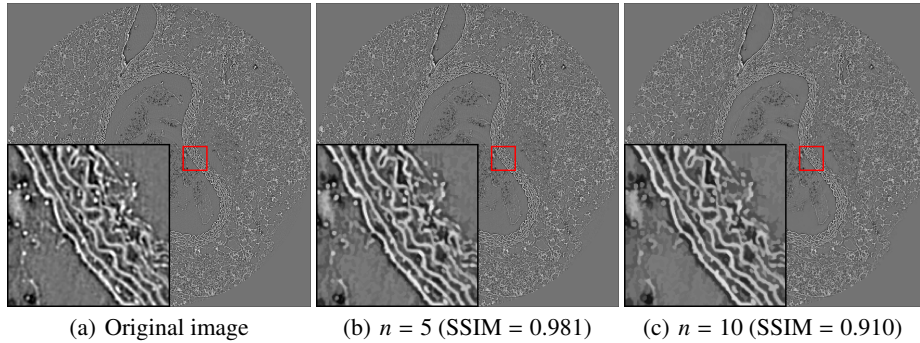


**Fig. 7.** (a) Remote sensing image (Landsat-7 image courtesy of the U.S. Geological Survey) ( $4.1 \cdot 10^5$  pixels, tree of shapes:  $1.7 \cdot 10^5$  nodes). (b,c) Simplified images obtained by area opening with (b)  $\lambda = 10$  (tree of shapes:  $1.5 \cdot 10^4$  nodes). (c)  $\lambda = 25$  (tree of shapes:  $9.7 \cdot 10^3$  nodes). The number of nodes / flat zones is progressively reduced with respect to  $\lambda$  in correlation with the SSIM decrease.

## 8 Conclusion

In this article, we introduced an approach for shifting nodes of a tree of shapes, ensuring that ( $P_1$ ) the resulting tree remains the tree of shapes of the image and ( $P_2$ ) the sign of the gradient of its contours is preserved. This opens the way to the development of a wide range of connected operators, designed to tackle specific issues in various applicative contexts.

The proposed framework is dedicated to the standard tree of shapes, that models grey-level images. On the one hand, the multivariate tree of shapes [7] was proposed for handling e.g. colour images. Extending our shifting paradigm to handle such multivariate images constitutes a perspective work. On the other hand, the topological tree of shapes [21] was recently introduced as a companion of the tree of shapes, which models the topology of grey-level images. Extending the proposed shifting approach to the topological tree of shapes is also a perspective work.



**Fig. 8.** (a) Biological image (synchrotron microtomography, SOLEIL ANATOMIX, project #20211303, courtesy S. Almagro) ( $4.2 \cdot 10^6$  pixels, tree of shapes:  $1.2 \cdot 10^6$  nodes). (b–c) Smoothed images after  $n$  iterations of mean filtering of the tree: (b)  $n = 5$  (tree of shapes:  $4.7 \cdot 10^5$  nodes), (c)  $n = 10$  (tree of shapes:  $2.3 \cdot 10^5$  nodes). The mean filter works by assigning to each node of the tree the mean value of its neighbouring nodes. The process is repeated  $n$  times. The number of nodes / flat zones is progressively reduced (removing noise) while the SSIM remains high (preserving structural information).

The ability of the proposed framework to simplify a tree of shapes, e.g. by removing (physical, semantic) noise and/or by reducing its combinatorial cost without losing significant information also opens the way to its involvement as a relevant image descriptor which could be embedded in deep-learning approaches, e.g. to model topological priors, or to ensure the preservation of structural properties of images, as already pioneered with the component tree [23].

## References

1. Baderot, J., Desvignes, M., Condat, L., Dalla Mura, M.: Tree of shapes cut for material segmentation guided by a design. In: ICASSP, Procs. pp. 2593–2597 (2020)
2. Ballester, C., Caselles, V., Igual, L.: Level lines selection with variational models for segmentation and encoding. *Journal of Mathematical Imaging and Vision* **27**, 5–27 (2006)
3. Boutry, N., Najman, L., Géraud, T.: Topological properties of the first non-local digitally well-composed interpolation on  $n$ -D cubical grids. *Journal of Mathematical Imaging and Vision* **62**, 1256–1284 (2020)
4. Breen, E.J., Jones, R.: Attribute openings, thinnings, and granulometries. *Computer Vision and Image Understanding* **64**, 377–389 (1996)
5. Carlinet, E., Crozet, S., Géraud, T.: The tree of shapes turned into a max-tree: A simple and efficient linear algorithm. In: ICIP, Procs. pp. 1488–1492 (2018)
6. Carlinet, E., Géraud, T.: A comparative review of component tree computation algorithms. *IEEE Transactions on Image Processing* **23**, 3885–3895 (2014)
7. Carlinet, E., Géraud, T.: MToS: A tree of shapes for multivariate images. *IEEE Transactions on Image Processing* **24**, 5330–5342 (2015)
8. Caselles, V., Monasse, P.: Grain filters. *Journal of Mathematical Imaging and Vision* **17**, 249–270 (2002)

9. Caselles, V., Meinhardt, E., Monasse, P.: Constructing the tree of shapes of an image by fusion of the trees of connected components of upper and lower level sets. *Positivity* **12**, 55–73 (2008)
10. Caselles, V., Monasse, P.: *Geometric Description of Images as Topographic Maps*. Lecture Notes in Mathematics, Springer (2010)
11. Crozet, S., Géraud, T.: A first parallel algorithm to compute the morphological tree of shapes of  $n$ D images. In: *ICIP, Procs.* pp. 2933–2937 (2014)
12. Géraud, T., Carlinet, E., Crozet, S., Najman, L.: A quasi-linear algorithm to compute the tree of shapes of  $n$ D. In: *ISMM, Procs.* pp. 98–110 (2013)
13. Jones, R.: Connected filtering and segmentation using component trees. *Computer Vision and Image Understanding* **75**, 215–228 (1999)
14. Kurtz, C., Naegel, B., Passat, N.: Connected filtering based on multivalued component-trees. *IEEE Transactions on Image Processing* **23**, 5152–5164 (2014)
15. Latecki, L.J., Eckhardt, U., Rosenfeld, A.: Well-composed sets. *Computer Vision and Image Understanding* **61**, 70–83 (1995)
16. Monasse, P.: A root-to-leaf algorithm computing the tree of shapes of an image. In: *RRPR, Procs.* pp. 43–54 (2018)
17. Monasse, P., Guichard, F.: Scale-space from a level lines tree. *Journal of Visual Communication and Image Representation* **11**, 224–236 (2000)
18. Najman, L., Schmitt, M.: Geodesic saliency of watershed contours and hierarchical segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**, 1163–1173 (1996)
19. Pan, Y., Birdwell, J.D., Djouadi, S.M.: Preferential image segmentation using trees of shapes. *IEEE Transactions on Image Processing* **18**, 854–866 (2009)
20. Passat, N., Kenmochi, Y.: A topological tree of shapes. In: *DGMM, Procs.* pp. 221–235 (2022)
21. Passat, N., Mendes Forte, J., Kenmochi, Y.: Morphological hierarchies: A unifying framework with new trees. *Journal of Mathematical Imaging and Vision* **65**, 718–753 (2023)
22. Perret, B., Chierchia, G., Cousty, J., Ferzoli Guimarães, S.J., Kenmochi, Y., Najman, L.: *Higra: Hierarchical graph analysis*. *SoftwareX* **10**, 100335 (2019)
23. Perret, B., Cousty, J.: Component tree loss function: Definition and optimization. In: *DGMM*. pp. 248–260 (2022)
24. Rosenfeld, A.: Adjacency in digital pictures. *Information and Control* **26**, 24–33 (1974)
25. Rosenfeld, A.: Digital topology. *The American Mathematical Monthly* **86**, 621–630 (1979)
26. Salembier, P., Garrido, L.: Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. *IEEE Transactions on Image Processing* **9**, 561–576 (2000)
27. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing* **7**, 555–570 (1998)
28. Salembier, P., Serra, J.: Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Transactions on Image Processing* **4**, 1153–1160 (1995)
29. Salembier, P., Wilkinson, M.H.F.: Connected operators. *IEEE Signal Processing Magazine* **26**, 136–157 (2009)
30. Song, Y., Zhang, A.: Monotonic tree. In: *DGCI, Procs.* pp. 114–123 (2002)
31. Tao, R., Qiao, J.: Fast component tree computation for images of limited levels. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**, 3059–3071 (2023)
32. Xu, Y., Géraud, T., Najman, L.: Connected filtering on tree-based shape-spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38**, 1126–1140 (2016)
33. Xu, Y., Géraud, T., Najman, L.: Hierarchical image simplification and segmentation based on mumford-shah-salient level line selection. *Pattern Recognition Letters* **83**, 278–286 (2016)