



HAL
open science

Connected Filtering on Tree-Based Shape-Spaces

Yongchao Xu, Thierry Géraud, Laurent Najman

► **To cite this version:**

Yongchao Xu, Thierry Géraud, Laurent Najman. Connected Filtering on Tree-Based Shape-Spaces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38 (6), pp.1126 - 1140. 10.1109/TPAMI.2015.2441070 . hal-01162437

HAL Id: hal-01162437

<https://hal.science/hal-01162437>

Submitted on 10 Jun 2015

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.

Connected Filtering on Tree-Based Shape-Spaces

Yongchao Xu, Thierry Géraud, Laurent Najman

Abstract—Connected filters are well-known for their good contour preservation property. A popular implementation strategy relies on tree-based image representations: for example, one can compute an attribute characterizing the connected component represented by each node of the tree and keep only the nodes for which the attribute is sufficiently high. This operation can be seen as a thresholding of the tree, seen as a graph whose nodes are weighted by the attribute. Rather than being satisfied with a mere thresholding, we propose to expand on this idea, and to apply connected filters on this latest graph. Consequently, the filtering is performed not in the space of the image, but in the space of shapes built from the image. Such a processing of shape-space filtering is a generalization of the existing tree-based connected operators. Indeed, the framework includes the classical existing connected operators by attributes. It also allows us to propose a class of novel connected operators from the leveling family, based on non-increasing attributes. Finally, we also propose a new class of connected operators that we call morphological *shapings*. Some illustrations and quantitative evaluations demonstrate the usefulness and robustness of the proposed shape-space filters.

Index Terms—Mathematical morphology, connected filtering, shape-space filtering, Max-tree, Min-tree, tree of shapes, graph, shape-based lower/upper leveling, blood vessel segmentation, shaping.

1 INTRODUCTION

MATHEMATICAL morphology, as originally developed by Matheron and Serra [1], proposes a set of morphological operators based on structuring elements. Later, Salembier and Serra [2], followed by Breen and Jones [3], proposed morphological operators based on attributes, rather than on elements. Such operators, also known as attribute filters or connected filters, have been popularized notably by Salembier, Wilkinson, and Ouzounis [4, 5, 6, 7, 8]. One popular implementation of such operators relies on transforming an image into an equivalent representation, namely a tree of components of the level sets of the image. Such trees are equivalent to the original image in the sense that the image can be reconstructed from its associated tree. Filtering then involves the design of a shape attribute that weighs how much a node of the tree fits a given shape. This tree-based implementation is depicted by the black path in Fig. 1.

Several approaches for filtering a tree of components (and hence the image) have been proposed. The more evolved approach consists in pruning the tree by removing some entire branches of the tree, an operation particularly relevant if the attribute is increasing on the tree (*i.e.*, if the attribute is always higher for the ancestors of a node). However, most shape attributes are not increasing. In this case, three pruning strategies have been

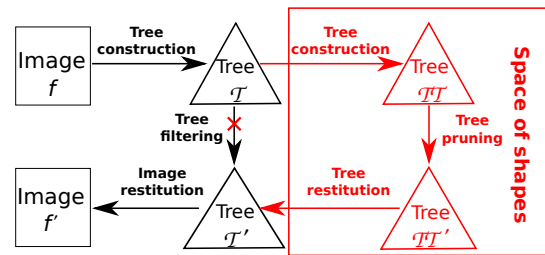


Fig. 1: Classical connected operators (black path) and the proposed shape-space filtering scheme (black+red path).

proposed (Min, Max, Viterbi; see Section 4.1 and [9, 4] for more details). They all choose a particular node on which to make a decision, and remove the subtree rooted at that node. Though it may give interesting results in some cases, it does not take into account the possibility that several relevant objects can have some inclusion relationship, meaning that they are on the same branch of the tree (*e.g.*, a ring object in a tree of shapes [10], see Fig. 6 (a)).

For non-increasing attributes, another commonly used approach is to simply remove the nodes of the tree for which the attribute is lower than a given threshold [9, 11, 4]. It is however often impossible to retrieve all expected objects with one unique (global) threshold. Fig. 2 shows the evolution of a shape attribute, the circularity [12], along two branches of the tree of shapes. The light round shape and the dark one are both meaningful round objects when compared to their context. However, their attribute values are very different. In order to obtain the light object, a high threshold is required, but then some non-desired shapes appear in the background in Fig. 2 (f).

This classical thresholding process can be seen as a simple filtering of the tree. In particular, it does not

- Yongchao Xu and Thierry Géraud are with EPITA Research and Development Laboratory (LRDE), 14-16 rue Voltaire, FR-94270 Le Kremlin-Bicêtre, France.
- Yongchao Xu, Thierry Géraud, and Laurent Najman are with the Laboratoire d'Informatique Gaspard-Monge, Université Paris-Est, Equipe A3SI, ESIEE Paris (FR-93160 Noisy-le-Grand, France).
- Yongchao Xu is with the Department of Signal and Image Processing, Telecom ParisTech, 46 rue Barrault, 75013 Paris, France.
E-mails: {yongchao.xu, thierry.geraud}@lrde.epita.fr, l.najman@esiee.fr

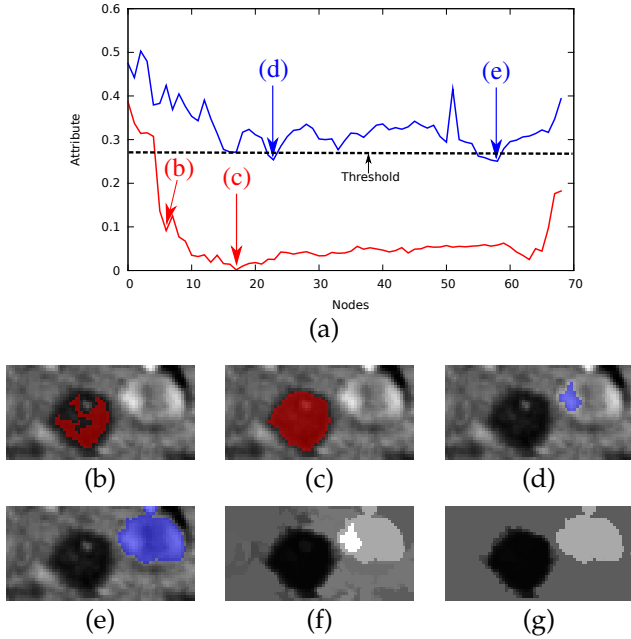


Fig. 2: (a) Evolution of the circularity attribute on two branches (containing respectively the red and blue shapes) of the tree of shapes [10]; (b to e) Some shapes; (f) Result of attribute thresholding; (g) Result of a shaping.

take into account the intrinsic parenthood relationship of the tree. The founding idea of this paper is to apply connected filters in the space of all the components of the image, such space being structured into a graph by the parenthood relationship (*i.e.*, the neighbors of a node are its children and its parent). This process is illustrated by the black+red path in Fig. 1. This surprising and simple idea has several deep consequences that were first exposed in a preliminary version of this study [13], where it is shown that this framework encompasses some usual attribute filtering operators. Novel connected filters based on non-increasing criteria can also be proposed. When the first tree \mathcal{T} is respectively a Min-tree or a Max-tree [9], such filters are novel morphological lower or upper levelings [14, 15] based on shapes. When the first tree \mathcal{T} is a tree of shapes [10], it gives birth to a novel family of connected filters that we call *morphological shapings*.

The main contributions of the present paper are: 1) an extensive discussion of the theory behind the framework of shape-space filtering; 2) the study of some theoretical properties of shape-space filters (*i.e.*, novel connected operators provided by this framework), morphological shapings, in particular, are also studied; and 3) some filtering illustrations allowig a qualitative comparison to classical thresholding-based methods, as well as an illustration of shape-based upper levelings to retinal blood vessel segmentation.

The rest of the paper is organized as follows. In Section 2 we briefly review a number of previous developments on which our methodology is based: connected operators and attribute filters. The theory of the frame-

work of shape-space filtering that we propose is detailed in Section 3. Section 4 is dedicated to demonstrate that this framework encompasses some classical attribute filters. Theoretical properties of various novel shape-space filters are explained in Section 5. Section 6 gives a set of illustrative experimental results using those shape-space filters. Finally we discuss and conclude in Section 7.

2 BACKGROUND

A discrete image is defined on a domain which can be seen as an undirected graph represented by a pair $G = (V, E)$, where V is the finite set of vertices and $E \subseteq (V \times V)$ is the set of edges. Each vertex $v \in V$ represents a pixel or a voxel of the image domain, and each edge $e \in E$ models the neighborhood relationship (classically, 4- or 8-connectivity for 2D images, and 6- or 26-connectivity in 3D images) between the two vertices composing e .

A graph (V, E) is said to be *connected* if, for any $x, y \in V$, there exists a path from x to y , which is a sequence of $n \geq 1$ vertices $(x_0 = x, x_1, \dots, x_n = y)$ such that every $x_i \in V$, and every $(x_i, x_{i+1}) \in E$. In this paper, the image domain V is connected.

A binary image X is a subset of the image domain that induces a subgraph (V_X, E_X) , such that V_X is the set of vertices representing the set of points of X , and $E_X = (V_X \times V_X) \cap E$. A binary set S is said to be connected if the subgraph (V_S, E_S) is connected. A *connected component* C of a binary image X is a connected subset of X with the maximal extent. This means for any C' such that $C \subseteq C' \subseteq X$, if C' is connected, then we have $C' = C$. More details about the notion of connectivity can be found in [16, 17, 7, 8].

Let $x \in V$. In the following, Γ_x denotes the operator that associates with a binary image X the connected component of X containing x if $x \in V_X$, otherwise \emptyset . Let Γ be the operator that associates the set of connected components with a binary image X . Formally, $\Gamma(X)$ denoted also by $\Gamma(V_X, E_X)$ is given by $\Gamma(X) = \{\Gamma_x(X), x \in V_X\}$.

In the sequel, we denote by \mathbb{F} the set of mappings $V \rightarrow \mathbb{R}$ or \mathbb{Z} . A grayscale image f is then an element of \mathbb{F} . Given a graph $G = (V, E)$, a pair (G, f) also denoted in the following by (V, E, f) is called a node-weighted graph, and it models an image f . In this paper, as a convention, operators on binary images are denoted with a capital Greek letter; the corresponding lowercase letter is used to denote the grayscale version.

2.1 Connected operators

In this section we briefly review the connected operators. More details can be found in [18, 2, 9, 4]. Generally speaking, such operators deal with a set of connected components $\mathbb{C} \subseteq \mathcal{P}(V)$ of an image, where \mathcal{P} denotes the set of all subsets of V . The connected operators act by filtering out some elements in \mathbb{C} and/or by changing their associated gray levels.

For a binary image X , the set \mathbb{C} is composed of two types of connected components: those in the foreground

$\Gamma(X)$ and those in the background $\Gamma(X^c)$, where X^c is the complement of X . We have $V_{X^c} = V \setminus V_X$. A *connected operator* Ψ working on X simply preserves or removes te connected components of X . This implies that the difference between X and $\Psi(X)$ is exclusively composed of the union of connected components of X and/or of its complement X^c .

For a grayscale image f , there are many possibilities for the set \mathbb{C} . The most trivial one is the set of elemental connected components called *flat zones* [2]. Let $L_h(V, f)$, also denoted simply by $L_h(f)$, be the set of points having a gray level h : $L_h(V, f) = \{x \in V \mid f(x) = h\}$. Then a connected component of the subgraph (L_h, E_{L_h}) is a flat zone, where $E_{L_h} = (L_h \times L_h) \cap E$. Indeed, for any point $x \in V$, $\Gamma_x(L_{f(x)}, E_{L_{f(x)}})$ is the flat zone containing that point. The set of flat zones denoted by \mathbb{C}_{FZ} of a grayscale image f is given by $\mathbb{C}_{FZ} = \mathbb{FZ}(V, E, f) = \{\Gamma_x(L_{f(x)}, E_{L_{f(x)}}), x \in V\}$, where $\mathbb{FZ}(V, E, f)$, also denoted simply by $\mathbb{FZ}(f)$, is the operator that gives the set of flat zones of a grayscale image f . Such a set of flat zones \mathbb{C}_{FZ} forms a partition of the image domain V called partition of flat zones [2]. The general definition of connected operators for grayscale images is based on this notion of partition of flat zones. An operator ψ on a grayscale image f is a *connected operator* if the partition of flat zones of $\psi(f)$ is coarser than the one of f , which means that $\forall C \in \mathbb{FZ}(f), \exists C' \in \mathbb{FZ}(\psi(f))$, such that $C \subseteq C' \Leftrightarrow \forall x \in V, \Gamma_x(L_{f(x)}, E_{L_{f(x)}}) \subseteq \Gamma_x(L_{\psi(f)(x)}, E_{L_{\psi(f)(x)}})$.

The flat zones of $\psi(f)$ are created by merging flat zones of f , implying that connected operators will not split the flat zones of f . Hence, connected operators do not introduce any new contour, and perfectly keep the location and shape of contours in the input image.

In practice, the flat zones in \mathbb{C}_{FZ} are usually individual points or small in size. Hence the decision whether or not to remove a connected components might be difficult to take. For that reason, other methods to create the set \mathbb{C} of a grayscale image have been proposed in the literature. A popular strategy is based on two types of threshold decomposition: upper level sets and lower level sets. For a given threshold value h , the upper level set $\mathcal{X}_h(V, f)$ or simply $\mathcal{X}_h(f)$ and the lower level set $\mathcal{X}^h(V, f)$ or simply $\mathcal{X}^h(f)$ are defined respectively by:

$$\mathcal{X}_h(V, f) = \{x \in V \mid f(x) \geq h\}, \quad (1)$$

$$\mathcal{X}^h(V, f) = \{x \in V \mid f(x) < h\}. \quad (2)$$

This leads to two different sets of connected components. They are given by the connected components of all possible upper (*resp.* lower) level sets denoted by $\mathbb{C}_>$ (*resp.* $\mathbb{C}^<$):

$$\mathbb{C}_> = \{\Gamma(\mathcal{X}_{f(x)}, E_{\mathcal{X}_{f(x)}}), x \in V\}, \quad (3)$$

$$\mathbb{C}^< = \{\Gamma(\mathcal{X}^{f(x)}, E_{\mathcal{X}^{f(x)}}), x \in V\}. \quad (4)$$

A third set of connected components is given by the fusion of $\mathbb{C}_>$ and $\mathbb{C}^<$ through the notion of *shapes* [10] (see [19, 20] for more details), where a shape is a connected component of an upper or lower level set with all cavities (*i.e.*, connected components of its complement)

filled in. We denote this third type of set of connected components by \mathbb{C}_\leq , it is given by:

$$\mathbb{C}_\leq = \{Sat(C), C \in \mathbb{C}_> \text{ or } C \in \mathbb{C}^<\}, \quad (5)$$

where *Sat* is the saturation operator that acts by filling the cavities (connected components of the background).

2.2 Attribute filters

Attribute filters [3, 9, 21] deal with connected components instead of individual points (as in the case of classical morphological operators originally developed by Matheron and Serra [1]). They act by preserving or by removing the connected components based on some attribute criterion. We denote by \mathbb{A} the set of mappings $(\mathbb{C}, \mathbb{F}) \rightarrow \mathbb{R}$ or \mathbb{Z} . An attribute function is then an element of \mathbb{A} that measures some interesting feature $\mathcal{A}(C, f)$ on each connected component C . When no confusion can occur, we use the notation $\mathcal{A}(C)$. In this paper, we distinguish the following attribute functions:

- An attribute function \mathcal{A} can be as simple as the gray level at which a connected component is obtained. We denote this specific function by \mathcal{F} , and we term $\mathcal{F}(C)$ the *level* of the connected component C . If the set of connected components $\mathbb{C}_>$ (*resp.* $\mathbb{C}^<$) is used, $\mathcal{F}(C)$ is the smallest (*resp.* largest) gray level inside each connected component C .
- An attribute function \mathcal{A} can also be more complicated, for example it can be some shape information measurement, such as circularity, compactness, or elongation [5]. Note that these attributes do not depend on the gray levels of the points inside the connected components, which means for any connected component $C \in \mathbb{C}$, and any two different mapping functions $f_1 \in \mathbb{F}$ and $f_2 \in \mathbb{F}$, $f_1 \neq f_2$, we always have $\mathcal{A}(C, f_1) = \mathcal{A}(C, f_2)$. We call such attribute functions *shape attributes*.
- Some attribute functions based on the gray levels of points inside the connected components might also be interesting, such as the height or some other attributes [12]. For a given connected component $C \in \mathbb{C}$, the height is given by $\text{height}(C, f) = \bigvee_{x \in C} f(x) - \bigwedge_{x \in C} f(x)$.

Let T be an attribute criterion: $\mathbb{C} \rightarrow \{\text{true}, \text{false}\}$. Typically, T is given by the comparison of the attribute function \mathcal{A} to a given threshold λ : $T(C) = (\mathcal{A}(C) \geq \lambda)$. Then the *trivial* attribute filter Γ_T on a connected component C returns the connected component C itself if $T(C)$ is true, and \emptyset otherwise, with the convention that $\Gamma_T(\emptyset) = \emptyset$. The binary anti-extensive attribute filter Γ^T for a binary image X is defined by $\Gamma^T(X) = \bigcup_{x \in X} \Gamma_T(\Gamma_x(X))$. Recall

that an operator Ψ on a binary image X is said to be anti-extensive if $\Psi(X) \subseteq X$. If \mathcal{A} is increasing, which means $C_1 \subseteq C_2 \Rightarrow \mathcal{A}(C_1) \leq \mathcal{A}(C_2)$, Γ^T is then an *attribute opening*, otherwise Γ^T is an *attribute thinning* [3].

The extension of the attribute filters to grayscale images is based on the set of connected components \mathbb{C} obtained from an image. The attribute filters on a grayscale image f consists in preserving or removing

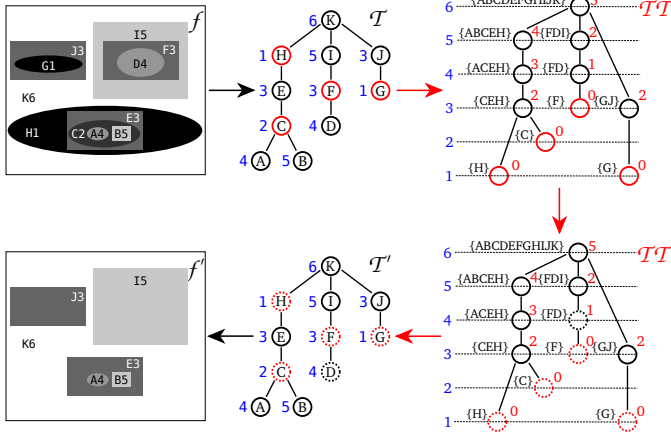


Fig. 3: An example of the workflow of shape-space filtering framework with $\mathcal{T}\mathcal{T}$ being the Min-tree of \mathcal{T} . Circles with a capital letter inside: nodes of \mathbb{C} ; Blue values: first attribute \mathcal{A} ; Circles without a letter inside: nodes of $\mathbb{C}\mathbb{C}$; Red values: second attribute $\mathcal{A}\mathcal{A}$; Dashed circles: filtered nodes with a given threshold value 2.

some connected components in \mathbb{C} . Let $\mathbb{C} = \mathbb{C}_>$ given by Eq. (3), the grayscale anti-extensive attribute filter γ^T for a given image f at any point x is defined by $\gamma^T(f)(x) = \bigvee \{h \mid x \in \Gamma^T(\mathcal{X}_h(f), E_{\mathcal{X}_h(f)})\}$. Recall that an operator ψ on a grayscale image f is said to be anti-extensive if $\psi(f) \leq f$.

The extensive attribute filters for binary images and grayscale images can be easily defined by the duality relationship with the anti-extensive versions defined above.

3 SHAPE-SPACE FILTERING

In the literature connected operators and attribute filters have generally been applied to the node-weighted graph (V, E, f) or shortly (G, f) of the image with $G = (V, E)$. They deal with the connected components of (G, f) and act by preserving or removing some of them in the set of connected components \mathbb{C} . We begin with studying some basic features of this set of connected components \mathbb{C} in Section 3.1. This leads to the introduction of the shape-space, a graph representation $G_{\mathbb{C}}$ in Section 3.2. Then we present in Section 3.3 the principle of the framework of shape-space filtering, which is, in short, applying the connected operators and attribute filters to the node-weighted graph $(G_{\mathbb{C}}, \mathcal{A})$. Finally, in Section 3.4, we detail how to use this framework efficiently, relying on tree-based image representations (as depicted in Fig. 1). A synthetic example of the scheme of shape-space filtering based on tree representation is given in Fig. 3.

3.1 The set of connected components \mathbb{C}

Any two different elements $C_i \in \mathbb{C}, C_j \in \mathbb{C}$ are either disjoint or nested: $\forall C_i \in \mathbb{C}, C_j \in \mathbb{C}, C_i \cap C_j \neq \emptyset \Rightarrow C_i \subseteq C_j$ or $C_j \subseteq C_i$. The set of connected components $\mathbb{C}_>, \mathbb{C}^<, \text{ and } \mathbb{C}_\leq^<$ given respectively by Eq. (3, 4, and 5) are such examples. The nesting property of these sets

of connected components is because $\forall h_1 \leq h_2, \mathcal{X}_{h_1} \supseteq \mathcal{X}_{h_2}$ and $\mathcal{X}^{h_1} \subseteq \mathcal{X}^{h_2}$. Note also that the data of the set of connected components $\mathbb{C}_>$ or $\mathbb{C}^<$ and the associated level function \mathcal{F} are sufficient to reconstruct the image thanks to the threshold superposition principle [22]: $f(x) = \bigvee \{h \mid x \in \mathcal{X}_h(f)\} = \bigwedge \{h \mid x \in \mathcal{X}^h(f)\}$. Let us denote this reconstruction operator by \mathbb{C}^{-1} . More generally, for any set of connected components \mathbb{C} obtained from some image f (whether it is $\mathbb{C}_>, \mathbb{C}^<, \text{ or } \mathbb{C}_\leq^<$) and its associated level function \mathcal{F} , the image f can be reconstructed by:

$$f(x) = \mathbb{C}^{-1}(x, \mathbb{C}, \mathcal{F}) = \mathcal{F}(C_x), \quad (6)$$

where C_x is the smallest connected component in \mathbb{C} such that $x \in C_x$, i.e., $C_x = \bigcap \{C_i \in \mathbb{C} \mid x \in C_i\}$. Thanks to this reconstruction principle, the set of connected components \mathbb{C} is an equivalent representation of the input image.

Classical connected operators and attribute filters remove connected components from the set \mathbb{C} (e.g., circles on the top-middle of Fig. 3), yielding a subset of connected components $\mathbb{C}' \subseteq \mathbb{C}$. Then, a filtered image f' is obtained from \mathbb{C}' based on the reconstruction principle in Eq. (6). Note that a slightly modified reconstruction rule has also been proposed in [11], see Section 4.1 for more details.

Such a classical filtering process is easy to apply if the attribute function is an increasing attribute \mathcal{A}^\uparrow : if a connected component $C_i \in \mathbb{C}$ is preserved, then any $C_j \in \mathbb{C}$ such that $C_i \subseteq C_j$ is also preserved. For a non-increasing attribute \mathcal{A} , several filtering rules have been proposed in the literature [4]. The simplest one among them is the thresholding-based filtering, where each connected component $C \in \mathbb{C}$ is analyzed individually. We note that the nesting property (represented by the lines linking any pair of circles on the top-middle of Fig. 3) of the family of connected components in \mathbb{C} is not taken into account in such a thresholding.

3.2 Shape-space

For any $C \in \mathbb{C}$, we denote by $Pr(C)$ the smallest element of \mathbb{C} containing C and different from C :

$$Pr(C) = \bigcap \{C_i \in \mathbb{C} \mid C \subsetneq C_i\}. \quad (7)$$

We observe that for any $C \in \mathbb{C}$, with C different from the image domain, $Pr(C) \neq \emptyset$. We also denote by $Pr^{-1}(C)$ the set of largest connected components contained in C and different from C , given by:

$$Pr^{-1}(C) = \{C_i \in \mathbb{C} \mid P(C_i) = C\}. \quad (8)$$

Note that $Pr^{-1}(C)$ might be empty. These two operators Pr and Pr^{-1} encode an adjacency relationship between the set of connected components in \mathbb{C} . We have $C_i = Pr(C_j) \Leftrightarrow C_j \in Pr^{-1}(C_i)$. Consequently, we can structure the family of connected components of \mathbb{C} into a graph representation denoted by $G_{\mathbb{C}} = (\mathbb{C}, E_{\mathbb{C}})$, where

$E_{\mathbb{C}}$ is given by Pr and Pr^{-1} :

$$E_{\mathbb{C}} = \{ (C_i, C_j) \in (\mathbb{C} \times \mathbb{C}) \mid \text{such as either } C_i = Pr(C_j) \\ \text{or } C_i \in Pr^{-1}(C_j) \}. \quad (9)$$

We call this graph representation $G_{\mathbb{C}}$ the *shape-space*. As compared to the image space, which is also modeled by a graph $G = (V, E)$, each node in $G_{\mathbb{C}}$ is a connected component of the image space. And each edge in $G_{\mathbb{C}}$ encodes the inclusion relationship of the connected components in \mathbb{C} . For example, the structure on the top-middle of Fig. 3 is a shape-space, where nodes are represented by circles, and edges by lines.

This shape-space has several interesting features. Firstly, it is invariant to contrast changes, that are increasing functions g applied to the image f , and covariant to continuous (topological) transformations. Besides, such a shape-space inherently embeds a morphological scale-space (the size of $Pr(C) \in \mathbb{C}$ is always larger than the size of $C \in \mathbb{C}$). Furthermore, this shape-space is an equivalent representation of an image, in the sense that the image can be reconstructed from the shape-space (thanks to Eq. (6) if \mathcal{F} is given). This shape-space also has another interesting property: contrary to scale-spaces, the contours of a given shape (topological boundaries of the connected components) correspond to actual contours in the image, without any blurring due to convolution with a kernel in the case of classical scale-space. Finally, this shape-space satisfies the *principle of causality* which is certainly the most fundamental principle of multi-scale analysis [23]. From this principle, for any pair of scales $\lambda_2 > \lambda_1$, the “structures” found at scale λ_2 should find a “cause” at scale λ_1 . Indeed, each connected component $C \in \mathbb{C}$ can be seen as a kind of “cause” of the connected component $Pr(C) \in \mathbb{C}$.

As an illustration of the usefulness of shape-spaces, we would like to mention two proposals of local feature detectors. The first proposal is the Maximally Stable Extremal Regions (MSER) [24, 25]. Note that MSER, broadly speaking, are equivalent to the Maximal Meaningful Lines previously proposed by Desolneux et al [26]. Indeed, a small variation in the components area occurs because of strong gradients in the image, which is the measure that is used in Desolneux et al.’s work to characterize meaningful level lines. The second proposal is the recent Tree-Based Morse Regions (TBMR) [27]; TBMR are truly invariant to contrast changes, a property which, combined with the other properties of shape-spaces, allows us to obtain state-of-the-art results in image registration and in multi-view 3D reconstruction.

3.3 Principle of shape-space filtering

For each element in the shape-space, the attribute function \mathcal{A} weighs how much that element fits a given criterion. The pair $(G_{\mathbb{C}}, \mathcal{A})$ is a node-weighted graph. In the classical use of attribute filters as described in Section 2.2, the filtering decision for each connected component $C \in \mathbb{C}$ is usually made upon the comparison of its attribute function $\mathcal{A}(C)$ to a given threshold value λ .

This thresholding process can be seen as a simple filtering of the graph $(G_{\mathbb{C}}, \mathcal{A})$. However, many sophisticated signal/image processing filters exist. Besides, it is well-known that a pertinent result with a unique thresholding value is usually difficult to obtain. Instead of just thresholding the graph $(G_{\mathbb{C}}, \mathcal{A})$, we propose to apply some signal/image filtering tools to the graph $(G_{\mathbb{C}}, \mathcal{A})$. Indeed, graphs are generic data structures that have been widely used in many scientific and engineering fields, notably for image analysis and computer vision [28, 29]. Since they are very flexible representations, a current trend is to use the classical tools for signal/image processing also for graphs [30]. Specifically, in this paper, we apply some connected operators to the graph $(G_{\mathbb{C}}, \mathcal{A})$. This process is the basis of the proposed framework.

Exactly as described in Section 2.1, in order to apply the connected operators to the graph $(G_{\mathbb{C}}, \mathcal{A})$, we create a set of connected components of the graph $(G_{\mathbb{C}}, \mathcal{A})$. We denote this second set of connected components by \mathbb{CC} , and we note that $\mathbb{CC} \subseteq \mathcal{P}(\mathbb{C})$. Each connected component $CC \in \mathbb{CC}$ is composed of a set of nested connected components $C \in \mathbb{C}$. We can also associate a second attribute \mathcal{AA} to each connected component $CC \in \mathbb{CC}$, where \mathcal{AA} is an element of the set of mappings $(\mathbb{CC}, \mathbb{A}) \rightarrow \mathbb{R}$ or \mathbb{Z} denoted by \mathbb{AA} . This second attribute function is usually designed to be increasing so that the filtering process is easy to apply. We filter out all the connected components $CC \in \mathbb{CC}$ such that $\mathcal{AA}(CC)$ is smaller than a given threshold λ . This filtering yields a subset of the connected components $\mathbb{CC}' \subseteq \mathbb{CC}$ given by:

$$\mathbb{CC}' = \{ CC \in \mathbb{CC} \mid \mathcal{AA}(CC) \geq \lambda \}. \quad (10)$$

For each connected component $C \in \mathbb{C}$ of the graph (G, f) , let $CC_C \in \mathbb{CC}$ be the smallest connected component of the graph $(G_{\mathbb{C}}, \mathcal{A})$ such that $C \in CC_C$:

$$CC_C = \bigcap \{ CC \in \mathbb{CC} \mid C \in CC \}. \quad (11)$$

Then the final filtered subset of connected components \mathbb{C}' of the graph (G, f) is given by:

$$\mathbb{C}' = \{ C \in \mathbb{C} \mid CC_C \in \mathbb{CC}' \}. \quad (12)$$

As described in Eq. (6), we can reconstruct an image f' from this subset of connected components \mathbb{C}' and its associated level function \mathcal{F} : $f' = \mathbb{C}^{-1}(\mathbb{C}', \mathcal{F})$. This reconstructed image f' is the final filtered image.

3.4 Shape-space filtering using trees

In practice, an efficient implementation of the connected operators and the framework of shape-space filtering rely on some tree-based representations (see [9, 10] for some more details). Indeed, if the whole domain of the graph G (*resp.* $G_{\mathbb{C}}$) belongs to the set of connected components $V \in \mathbb{C}$ (*resp.* $\mathbb{C} \in \mathbb{CC}$), then the set of connected components \mathbb{C} (*resp.* \mathbb{CC}) can be organized into a tree structure denoted by \mathcal{T} (*resp.* \mathcal{TT}) thanks to the nesting property. Each node of the tree \mathcal{T} (*resp.* \mathcal{TT}) corresponds to a connected component $C \in \mathbb{C}$ (*resp.* $CC \in \mathbb{CC}$). The parenthood relationship of the tree \mathcal{T}

(resp. \mathcal{TT}) is given by Pr defined in Eq (7) (resp. PPr which is similar to Pr but defined on \mathbb{CC}). The node $Pr(C) \in \mathcal{T}$ (resp. $PPr(CC) \in \mathcal{TT}$) is said to be the *parent* of the node $C \in \mathcal{T}$ (resp. $CC \in \mathcal{TT}$), and C (resp. CC) is said to be a *child* of $Pr(C)$ (resp. $PPr(CC)$). Note that the node representing $V \in \mathbb{C}$ or $\mathbb{C} \in \mathbb{CC}$ has no parent. The set of connected components $\mathbb{C}_>$ in Eq. (3) and $\mathbb{C}_<$ in Eq. (4) are organized into respectively Max-tree and Min-tree representation [9]. And the set of connected components \mathbb{C}_\leq in Eq. (5) gives the tree of shapes [10]. Using these tree-based representations, we call the shape-space $G_{\mathbb{C}} = (\mathbb{C}, E_{\mathbb{C}})$ described in Section 3.2 the *tree-based shape-space*. We also denote it by $G_{\mathcal{T}}$. Each node of the graph $G_{\mathcal{T}}$ is adjacent to its parent, so each node has its parent and its children as its neighbors.

A schematic overview of the framework of shape-space filtering relying on the tree-based representations is depicted by the black+red path in Fig. 1. The filtering is performed on the second tree \mathcal{TT} , that contrasts with classical connected operators which filter the first tree \mathcal{T} . The whole process is composed of: 1) The first tree \mathcal{T} construction to create \mathbb{C} of graph (G, f) ; 2) The second tree \mathcal{TT} construction to create \mathbb{CC} of graph $(G_{\mathbb{C}}, \mathcal{A})$; 3) The second tree pruning that yields a subset $\mathbb{CC}' \subseteq \mathbb{CC}$ contained in the simplified tree \mathcal{TT}' ; 4) A simplified tree \mathcal{T}' restitution to produce the filtered subset $\mathbb{C}' \subseteq \mathbb{C}$, and an image reconstruction from \mathbb{C}' . Fig. 3 depicts a synthetic example of the proposed framework using tree representation. We hereafter rely on this example to detail each step of the proposed framework.

3.4.1 Creation of \mathbb{C} by tree representation \mathcal{T}

First of all, we build a tree representation of graph (G, f) to create the set of connected components \mathbb{C} . The choice of the type of tree \mathcal{T} depends on the targeted application: the shapes (objects) of interest have to be present at some nodes of the chosen tree. The tree of shapes is used in Fig. 3. During the tree \mathcal{T} construction, we are able to compute incrementally a lot of information, based on which the attribute \mathcal{A} (e.g., area, gray level \mathcal{F} , or some more evolved shape attributes) is derived. The pair $(G_{\mathcal{T}}, \mathcal{A})$ is a node-weighted graph modeling the tree-based shape space. The graph $(G_{\mathcal{T}}, \mathcal{A})$ in Fig. 3 has four regional minima (represented by red circles).

3.4.2 Construction of \mathbb{CC} by the second tree \mathcal{TT}

As we have done in the previous step for the graph (G, f) , we construct the set of connected components \mathbb{CC} of graph $(G_{\mathcal{T}}, \mathcal{A})$ by constructing a tree representation \mathcal{TT} . In this paper, this second tree \mathcal{TT} is either a Min-tree or a Max-tree representation. The choice between them depends on the application and on the nature of the attribute \mathcal{A} . Indeed, the attribute \mathcal{A} encodes the probability for a shape to be of a given type. If we want to filter out the shapes different from that type, we use a Min-tree for \mathcal{TT} , so that the minima of the space of shapes are the less probable shapes. In other words, for a node of \mathcal{T} to be a leaf of \mathcal{TT} , its parent and its children have to be more probable than the node itself.

Conversely, if we want to preserve the shapes of a given type, then we use a Max-Tree for \mathcal{TT} .

Two different “objects” located in the same branch of \mathcal{T} are now possibly present in two different branches of \mathcal{TT} . For example, in Fig. 3, nodes C and H of \mathcal{T} are now in two distinct branches of \mathcal{TT} (Min-tree of \mathcal{T}). Hence, these two “objects” can be extracted separately, which is difficult to achieve using only \mathcal{T} . In this example, the four regional minima of the graph $(G_{\mathcal{T}}, \mathcal{A})$ are represented by the four leaves (red circles) of \mathcal{TT} .

3.4.3 Filtering of \mathcal{TT}

A tree filtering of \mathcal{TT} is performed based on a second attribute \mathcal{AA} . This attribute \mathcal{AA} is always an increasing attribute in order to ensure that the second tree filtering can be achieved with a simple pruning strategy. The design of this second attribute \mathcal{AA} is quite flexible. Usually, it can also be computed incrementally during \mathcal{TT} construction, based on the first attribute \mathcal{A} (for instance, the height of \mathcal{A}), or based on the contextual information on the image domain around the shapes that CC represents.

The pruning is then based on comparing \mathcal{AA} to a given threshold. Let us remark that depending on the application, two different pruning strategies can be used. For the purpose of filtering out some non-desired shapes, the nodes to be pruned are the subtrees rooted just above the leaves. If we want to select the shapes corresponding to the desired shapes (represented by the leaves of \mathcal{TT}), the pruning strategy removes the nodes that are close to the root node of \mathcal{TT} . This is equivalent to preservation of subtrees containing the leaves.

In Fig. 3, the second tree is filtered by pruning the nodes CC whose attribute value \mathcal{AA} , e.g., the height of \mathcal{A} applied on shape-space, is less than 2. This is an example of filtering out the non-desired shapes around the leaves.

3.4.4 Tree restitution and image restitution

The tree restitution step is trivial. The simplified tree \mathcal{T}' is reconstructed by removing the set of nodes $\{C_i\}_i$ contained in the series of filtered nodes $\{CC_k\}_k$, and by updating the parenthood relationship. For example, the simplified tree \mathcal{T}' in Fig. 3 is obtained by removing the nodes (dashed circles) contained in the set of filtered nodes of \mathcal{TT}' , and each preserved node in \mathcal{T} takes its lowest preserved ancestor as its parent.

For any point $x \in V$, let $C_x^a \in \mathcal{T}'$ be the lowest preserved ancestor node that contains the point x . Then based on the principle of image reconstruction described in Section 3.1, the filtered image f' at any point x is given by:

$$f'(x) = \begin{cases} \mathcal{F}(C_x) & \text{if } C_x \in \mathcal{T}', \\ \mathcal{F}(C_x^a) & \text{otherwise.} \end{cases} \quad (13)$$

A similar reconstruction strategy by maintaining the local contrast of the preserved nodes is detailed in Section 4.1.

Note that in Fig. 3, the nodes C and H are removed, while the node E lying between them in the same branch is preserved. None of the existing pruning strategies

described in Section 4.1 can achieve such a result. Indeed, the nodes E and J with $\mathcal{A} = 3$ are preserved, while the node F with $\mathcal{A} = 3$ and even the node D with $\mathcal{A} = 4$ are filtered out. Such a behavior cannot be obtained with a threshold-based strategy.

4 REVISITING CLASSICAL ATTRIBUTE-BASED CONNECTED OPERATORS

The framework of shape-space filtering deals with connected components which are composed of flat zones. This guarantees that the operators of the proposed framework belong to the class of connected operators. In this section, we show in Section 4.2 that the framework of shape-space filtering encompasses some classical strategies of attribute filters which are detailed in Section 4.1.

4.1 Classical rules for tree filtering

Filtering the tree \mathcal{T} with some increasing attributes \mathcal{A}^\uparrow is rather straightforward: it is simply performed by removing all the nodes that do not satisfy the attribute criterion T . As the attribute \mathcal{A}^\uparrow is increasing, it amounts to removing a branch of the tree: if a node does not satisfy T , then none of its descendants satisfies it. The pruned points take the respective gray levels of their respective lowest preserved ancestors. For non-increasing attributes, which is the usual case for many shape attributes, there is no straightforward strategy to filter the tree. Salembier *et al.* [9, 4] propose three pruning strategies (*Min*, *Max*, *Viterbi*) and a threshold-based rule (*Direct*). Urbach *et al.* [11] also propose a threshold-based strategy (*Subtractive*). The decisions of these rules for a given attribute threshold λ are described as follows:

- *Min*: a node C is removed if $\mathcal{A}(C) < \lambda$ or if there exists one of its ancestors C_a such that $\mathcal{A}(C_a) < \lambda$.
- *Max*: a node C is removed if $\mathcal{A}(C) < \lambda$ and for all its descendants C_d , $\mathcal{A}(C_d) < \lambda$ holds.
- *Viterbi*: removal and preservation of nodes are determined by a cost optimization process. Each transition is assigned with a cost. For each leaf node, the branch with the lowest cost to the root node is taken. See [9] for more details.
- *Direct*: a node C is removed if $\mathcal{A}(C) < \lambda$. The pixels stored in C but not in any of its descendants take the gray level of the first preserved ancestor starting from C to the root.
- *Subtractive*: as the direct rule, but it differs in image reconstruction. The change in the gray level of the modified node is applied to all its descendants, so that the contrast with the local background remains unchanged.

Let us notice that all those rules have some drawbacks. Pruning strategies can not deal with the case where two interesting objects are present in the same branch. With the two threshold-based rules, it is often impossible to retrieve at the same time all the expected objects with one unique global threshold.

4.2 Encompassing some classical attribute filters

We now explore the effect on the image of filtering \mathcal{TT} , the Min-tree of the node-weighted graph $(G_{\mathcal{T}}, \mathcal{A})$, with some specific attributes. We show that those attributes allow us to retrieve the classical filtering rules described in the previous section.

Case A: Pruning of increasing attribute \mathcal{A}

In the most trivial case, the attribute \mathcal{A} is increasing, and the classical connected filters are equivalent to a pruning of the tree.

Proposition 1: For any tree representation \mathcal{T} , if \mathcal{A} is strictly increasing, let \mathcal{TT} be the Min-tree of the node-weighted graph $(G_{\mathcal{T}}, \mathcal{A})$, then \mathcal{TT} is isomorphic to \mathcal{T} .

Proof: Since \mathcal{A} is strictly increasing, so for any given node C , $\mathcal{A}(C) < \mathcal{A}(C_a)$ holds for any ancestor node C_a of C , which means that the leaves of \mathcal{T} are regional minima of graph $(G_{\mathcal{T}}, \mathcal{A})$. These regional minima lie also on the leaves of \mathcal{TT} being the Min-tree. Furthermore, for any pair of neighboring nodes $(C_i, C_j) \in E_{\mathcal{C}}$, either $C_i = P(C_j)$ or $C_i \in P^{-1}(C_j)$. Suppose that the former one holds, then $C_j \subset C_i$, $\mathcal{A}(C_j) < \mathcal{A}(C_i)$. Let $C_k \in \mathbb{C}$, $C_k \neq C_i$ be any neighboring node of C_j , since the parent of each node on a tree structure is unique, so C_k is a child of C_j , which means $\mathcal{A}(C_k) < \mathcal{A}(C_j)$. In consequence, we have $CC_{C_i} = PPr(CC_{C_j})$. So \mathcal{TT} has the same structure of \mathcal{T} .

Let \mathcal{AA} be the current level of the second tree \mathcal{TT} , which means $\forall C \in \mathbb{C}$, $\mathcal{AA}(CC_C) = \mathcal{A}(C)$. Pruning \mathcal{TT} is equivalent to pruning \mathcal{T} . In other words, shape-space filtering encompasses the classical filtering strategy in this case, but we do not have to test whether the attribute \mathcal{A} is increasing or not.

Case B: Thresholding of non-increasing attribute \mathcal{A}

A shape attribute \mathcal{A} is more often non-increasing. In such a case, there exists some pair of vertices $(C_i, C_j) \in E_{\mathbb{C}}$, such that $C_j = P(C_i)$, and $\mathcal{A}(C_j) < \mathcal{A}(C_i)$. For example, let \mathcal{TT} be the Min-tree of the graph $(G_{\mathcal{T}}, \mathcal{A})$, then the node $CC_{C_i} \in \mathbb{CC}$ is hence an ancestor of $CC_{C_j} \in \mathbb{CC}$ on \mathcal{TT} . So \mathcal{TT} is different from \mathcal{T} . Furthermore, just like the increasing attribute case, let \mathcal{AA} be the current level of \mathcal{TT} , we have $\mathcal{AA}(CC_C) = \mathcal{A}(C)$ for any $C \in \mathbb{C}$. Pruning \mathcal{TT} by \mathcal{AA} is equivalent to thresholding \mathcal{T} . Accordingly, shape-space filtering encompasses the threshold-based strategies (*Direct* and *Subtractive* rules), which are context-free or adjacency stable operators, as explained in [31]. Let us remark that, although shape-space filtering theoretically encompasses the cases presented above, a direct implementation (not relying on the proposed framework) is more efficient. Let us also remark that it is impossible (without a global rule) to retrieve in the proposed framework the contextual filtering [32] pruning strategies described in Section 4.1, based on *Min*, *Max*, and *Viterbi*.

The second attribute \mathcal{AA} can be different from \mathcal{A} . For example, it can be any measure based on \mathcal{A} or even some new attribute/measure computed from the image domain (*e.g.*, the total variation inside the context region represented by the node CC). This is when shape-space filtering becomes particularly useful.

5 SOME NOVEL CONNECTED OPERATORS

Shape-space filtering is more flexible than conventional connected operators and offers some new possibilities. In this section, we introduce two novel classes of connected operators: shape-based lower/upper levelings in Section 5.1 and morphological shapings in Section 5.2.

In the rest of this paper, we study the following process:

- we first build a family \mathbb{C} of the connected components of level sets of the image. In other words, \mathbb{C} is made of the connected components corresponding to the nodes of either the Min-tree, the Max-tree or the tree of shapes;
- using the parenthood relationship in Eq. (7), we build the graph $(\mathbb{C}, E_{\mathbb{C}})$, where $E_{\mathbb{C}}$ is given by Eq. (9);
- we choose an attribute \mathcal{A} that evaluates any element of \mathbb{C} . This attribute does not need any particular property, other than the ones needed for the application at hand;
- we choose an attribute \mathcal{AA} , increasing for the connected components of node-weighted graph $(\mathbb{C}, E_{\mathbb{C}}, \mathcal{A})$. This attribute \mathcal{AA} evaluates the strength of the attribute \mathcal{A} in the graph $(\mathbb{C}, E_{\mathbb{C}})$;
- we then make an attribute filtering on the node-weighted graph $(\mathbb{C}, E_{\mathbb{C}}, \mathcal{A})$ with the attribute \mathcal{AA} .

We call such an attribute-based connected operator a *shape-space filter*. Note that this definition is not to be confused with the definition of *shape filter* given in [11]: specifically, a shape filter is scale, rotation and translation invariant.

5.1 Shape-based lower/upper levelings

Let us now detail the first type of novel connected operators offered by the framework of shape-space filtering, in the case of \mathcal{T} being a Max-tree or Min-tree representation of the image f . For that, we first recall the notion of levelings [14].

The levelings form a subclass of connected operators. They act by enlarging the flat zones by suppressing many details, while keeping the sharpness of the transition zones. The levelings are the intersection of two subclasses: the lower levelings and upper levelings.

Definition 1: An operator ψ is a *lower leveling* of a grayscale image f if and only if for any pair of neighboring vertices $(x, y) : \psi(f)(x) > \psi(f)(y) \Rightarrow \psi(f)(y) \geq f(y)$.

Definition 2: An operator ψ is an *upper leveling* of a grayscale image f if and only if for any pair of neighboring vertices $(x, y) : \psi(f)(x) > \psi(f)(y) \Rightarrow \psi(f)(x) \leq f(x)$.

Definition 3: An operator ψ is a *leveling* of a grayscale image f if and only if for any pair of neighboring points $(x, y) : \psi(f)(x) > \psi(f)(y) \Rightarrow f(x) \geq \psi(f)(x)$ and $\psi(f)(y) \geq f(y)$.

The definition 3 states that if there is a transition in the output image after leveling, the transition exists in the initial image. This is because $\psi(f)(x) >$

$\psi(f)(y) \Rightarrow f(x) \geq \psi(f)(x) > \psi(f)(y) \geq f(y)$. Furthermore, the interval of the transition in the output image $[\psi(f)(y), \psi(f)(x)]$ is contained in the interval of the transition in the input image $[f(y), f(x)]$.

The readers can refer to [14, 15] for more details about the properties of levelings.

Proposition 2: If \mathbb{C} is made from a Max-tree \mathcal{T} , and if the attribute function is a (non-increasing) attribute, any shape-space filter is an upper leveling. We term such an operator $\psi_{s\uparrow}$ a *shape-based upper leveling*.

Proof: Let \mathcal{T} be a Max-tree representation, then no matter what type of tree \mathcal{TT} is and no matter how \mathcal{TT} is pruned, the simplified tree \mathcal{T}' has always a Max-tree structure in the sense that gray level for the ancestors is always lowered. In the image reconstruction step, the pixels stored in some removed node C_r take the gray level of the first preserved ancestor C_a (*Direct rule*) or even lowered with the change induced by the removed ancestors (*Subtractive rule*). Hence, $\forall x \in V, \psi_{s\uparrow}(f)(x) \leq f(x)$ always holds. By Definition 2, such an operator $\psi_{s\uparrow}$ is an upper leveling.

Proposition 3: If \mathbb{C} is made from a Min-tree \mathcal{T} , and if the attribute function is a (non-increasing) attribute, any shape-space filter is a lower leveling. We term such an operator $\psi_{s\downarrow}$ a *shape-based lower leveling*.

Proof: Let \mathcal{T} be a Min-tree, the simplified tree \mathcal{T}' is still a Min-tree in the sense that the gray levels of the ancestors are always higher. So, similarly to the proof of Prop. 2, $\forall x \in V, \psi_{s\downarrow}(f)(x) \geq f(x)$ holds. Based on Definition 1, such an operator $\psi_{s\downarrow}$ is a lower leveling.

When the first tree \mathcal{T} is a Max-tree (*resp* Min-tree), the reason why shape-space filtering gives shape-based upper (*resp*. lower) leveling is that any anti-extensive (*resp*. extensive) operator is an upper (*resp*. lower) leveling. Let us also remark that in both cases, the operator ψ_s is not a leveling. Indeed, for any pair of points $(x, y) \in E$, we have either $f(x) \geq \psi_s(f)(x)$ and $f(y) \geq \psi_s(f)(y)$, or $\psi_s(f)(x) \geq f(x)$ and $\psi_s(f)(y) \geq f(y)$. Therefore, the operator ψ_s is not a leveling (except when $\psi_s(f)$ is a constant), according to Definition 3.

A classical upper (*resp*. lower) leveling removes some details around the regional minima (*resp*. maxima). In practice, this is equivalent to pruning the Min-tree (*resp*. Max-tree) with an increasing attribute \mathcal{A}^\uparrow . Nevertheless, a shape-based upper (*resp*. lower) leveling is based on some connected filtering on the shape-space built from the Max-tree (*resp*. Min-tree) representation. It filters out some details of unwanted bright (*resp*. dark) shapes on the basis of a user defined non-increasing attributes.

Recall that an operator ψ is said to be *idempotent* if and only if $\psi(\psi(f)) = \psi(f)$.

Proposition 4: If the second tree \mathcal{TT} filtering is idempotent then the shape-based upper leveling $\psi_{s\uparrow}$ and lower-leveling $\psi_{s\downarrow}$ are idempotent.

Proof: Let $\mathcal{T}_0, \mathcal{TT}_0, \mathcal{TT}'_0$, and \mathcal{T}'_0 be the tree structures corresponding to the shape-based morphological operator ψ_s applied to f . $\mathcal{T}_1, \mathcal{TT}_1, \mathcal{TT}'_1$, and \mathcal{T}'_1 are the tree structures corresponding to ψ_s applied to $f' = \psi_s(f)$. As \mathcal{T}'_0 and \mathcal{T}'_1 are either a Max-tree or a Min-tree, it is trivial to see that $\mathcal{T}'_1 = \mathcal{T}'_0$, thanks to the strict fixed order of the

gray level between neighboring nodes of those trees. As the second tree filtering is based on the pruning strategy, it is equivalent to removing some blobs around minima or maxima of the graph $(G_{\mathcal{T}_0}, \mathcal{A})$. As a consequence, the second tree of graph $(G_{\mathcal{T}'_0}, \mathcal{A})$ is the same as \mathcal{T}'_0 , which means that $\mathcal{T}\mathcal{T}'_1 = \mathcal{T}\mathcal{T}'_0$. The idempotent second tree filtering yields: $\mathcal{T}\mathcal{T}'_1 = \mathcal{T}\mathcal{T}'_0 \Rightarrow \mathcal{T}'_1 = \mathcal{T}'_0$, so $\psi_s(\psi_s(f)) = \psi_s(f)$ holds.

5.2 Morphological shapings

Unlike the shape-based upper or lower levelings, which deals only with bright or dark shapes, we introduce in this section a second type of connected operators which process both bright and dark shapes at the same time.

Proposition 5: If \mathbb{C} is made from the tree of shapes \mathcal{T} , and if \mathcal{A} is a non-increasing attribute, the operator ψ_s given by the shape-space filtering is in general not a leveling.

Proof: Since \mathcal{A} is non-increasing, and any shape-space operator gives a filtered tree \mathcal{T}' which does not result from a pruning of \mathcal{T} , a pair of neighboring vertices (x, y) can exist, such that $C_x \subset C_y$, and C_y is removed while C_x is preserved. Let C_z be the lowest preserved ancestor of C_y . Then we have $\psi_s(f)(x) = f(x)$, $\psi_s(f)(y) = f(z)$. However, as the tree \mathcal{T} is the tree of shapes, the order of the gray level of a node and any of its ancestors is not monotonous *a priori*. So it is possible that $f(z) < f(x)$ and $f(z) < f(y)$. In this case, we have $\psi_s(f)(x) > \psi_s(f)(y)$, $f(x) \geq \psi_s(f)(x)$ and $\psi_s(f)(y) < f(y)$. This is in contradiction with the definition of a leveling (see Definition (3)).

Definition 4: If \mathbb{C} is made from the connected components corresponding to the tree of shapes \mathcal{T} , and if \mathcal{A} is a non-increasing attribute, then the associated shape-space filter is called a *morphological shaping*.

In the sequel, the symbol \mathcal{S} is used for denoting a morphological shaping.

Proposition 6: A morphological shaping \mathcal{S} with shape attribute \mathcal{A} is a self-dual operator.

Proof: Since the tree of shapes \mathcal{T} is self-dual, the tree of shapes of graph (G, f) denoted by \mathcal{T}^+ has the same structure as the one of graph $(G, -f)$ denoted by \mathcal{T}^- . Consequently, the set of connected components made from \mathcal{T}^+ denoted as \mathbb{C}^+ is equal to the one made from \mathcal{T}^- denoted as \mathbb{C}^- . By definition, the shape attribute \mathcal{A} does not depend on the gray levels of the points inside the connected components, which means for any $C \in \mathbb{C}^+$, $C \in \mathbb{C}^-$, we have $\mathcal{A}(C, f) = \mathcal{A}(C, -f)$. In consequence, the graph $(G_{\mathcal{T}^+}, \mathcal{A})$ is equal to the graph $(G_{\mathcal{T}^-}, \mathcal{A})$. Then we have $\mathcal{T}\mathcal{T}^+ = \mathcal{T}\mathcal{T}^- \Rightarrow \mathcal{T}'^+ = \mathcal{T}'^-$ and $f' = -f'$, which means that $\mathcal{S}(-f) = -\mathcal{S}(f)$. Thus the shaping \mathcal{S} is a self-dual operator.

To make a shaping \mathcal{S} idempotent, the first condition to satisfy is that the tree of shapes of $\mathcal{S}(f)$ denoted by \mathcal{T}_1 is equivalent to the simplified tree \mathcal{T}'_0 from which the filtered image $\mathcal{S}(f)$ is reconstructed. However, this is not a trivial requirement due to the unfixed order of gray levels of the neighboring nodes on the tree of shapes \mathcal{T} . A node $C_k \in \mathbb{C}$ of \mathcal{T} is obtained by filling the cavities of

either a component of a upper level set (see Eq. (1)), or a component of a lower level set (see Eq. (2)). Let us denote the family of these two types of nodes respectively by $\mathbb{C}_>$ and $\mathbb{C}_<$. We have $\mathbb{C} = \mathbb{C}_> \cup \mathbb{C}_<$.

Proposition 7: Let \mathcal{T} be the tree of shapes, \mathbb{C}' be the set of preserved connected components, and P' be the parenthood relationship between connected components in \mathbb{C}' . For each preserved node $C \in \mathbb{C}'$, $C' \subseteq \mathbb{C}$, if $\mathcal{F}'(C) < \mathcal{F}'(P'(C))$ holds for $C \in \mathbb{C}_<$ of \mathcal{T} , and $\mathcal{F}'(C) > \mathcal{F}'(P'(C))$ holds for $C \in \mathbb{C}_>$ of \mathcal{T} , then the tree \mathcal{T}_1 constructed from f' is equal to \mathcal{T}' from which f' is reconstructed.

Proof: Thresholding the reconstructed image f' by Eq. (1) and Eq. (2) yields some shapes which can also be found in \mathcal{T}' . This correspondence is guaranteed by the condition $\forall C \in \mathbb{C}_< \text{ remained in } \mathcal{T}', \mathcal{F}'(C) < \mathcal{F}'(P'(C))$ and $\forall C \in \mathbb{C}_> \text{ remained in } \mathcal{T}', \mathcal{F}'(C) > \mathcal{F}'(P'(C))$. Based on the definition of the tree of shapes [10], the inclusion relationship between those shapes yields a unique tree, which is equal to \mathcal{T}' .

Proposition 8: If the second tree filtering $\mathcal{T}\mathcal{T}$ is idempotent and if the type (either a $\mathbb{C}_>$ or $\mathbb{C}_<$ node) of the preserved nodes does not change, then the shaping \mathcal{S} is idempotent.

Proof: See Prop. 7 and the proof of Prop. 4.

6 ILLUSTRATIONS AND EXPERIMENTS

In this section, we present some illustrations and experimental results based on the novel connected operators introduced in the previous section. First of all, in Section 6.1, we provide several filtering results of some shape-space filters, and we compare them, from a qualitative point of view, to those based on thresholding strategies. This qualitative comparison demonstrates the robustness and flexibility of the proposed framework. Then, in Section 6.2, we evaluate quantitatively one of the shape-space filters against its corresponding threshold-based filter, in the context of retinal blood vessel segmentation.

6.1 Comparison with classical approaches

We compare several shape-space filters to their corresponding classical attribute filters based on thresholding strategies. First of all, we illustrate two examples of shape-based lower levelings. In Fig. 4, we aim to preserve the round disks while filtering out the other structures in Fig. 4 (a). The first tree representation is a Min-tree, and we use for the attribute \mathcal{A} the inverse of the compactness [5, 11], the compactness being defined by $L^2 / (4\pi A)$, where A and L are respectively the area and the perimeter of the component. Fig. 4 (c) is the result of a subtractive reconstruction and Fig. 4 (d) is obtained by taking the average value for the pixels inside each preserved node excluding its preserved descendants. We observe that, with a low thresholding value, some unwanted objects remain. When we augment the thresholding value (shown in Fig. 4 (e) and (f)), some of the expected objects are filtered out. At the same time,

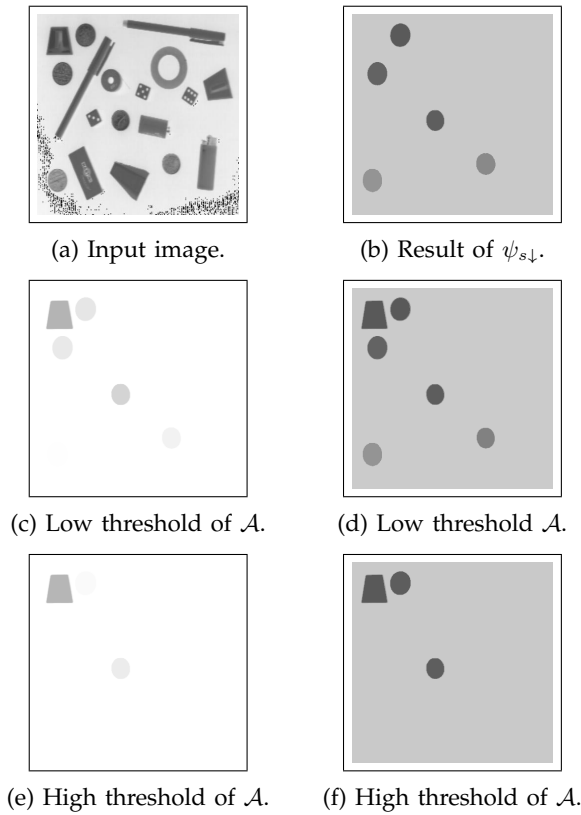


Fig. 4: Comparison between shape-based lower leveling and attribute thresholding, with \mathcal{A} being the compactness. Images (c,e) are obtained with the subtractive rule and images (d, f) with the average [see text].

some unexpected objects are still preserved. The result of a shape-based lower leveling is shown in Fig. 4 (b): the second tree $\mathcal{T}\mathcal{T}$ is a Min-tree, and the second attribute function $\mathcal{A}\mathcal{A}$ is the combination of the attribute \mathcal{A} itself with the total variation inside the context region represented by the node CC . We observe that all the five round disks are preserved while all the other structures have been filtered out. In Fig. 5, we use the average of the gradient magnitude on the shape contour as attribute \mathcal{A} . As illustrated in Fig. 5 (c) and (d), thresholding this attribute with a low value preserves some structures (inside the red boxes) in the background. To get rid of these unexpected structures, a higher thresholding value is applied. But, as shown in Fig. 5 (e) and (f), some of the expected structures (inside the blue boxes) are also filtered out. The result of the shape-based lower leveling (Fig. 5 (b)) is cleaner. We preserve any node in the second tree $\mathcal{T}\mathcal{T}$ (a Max-tree) with a small height, and if furthermore the node contains a maximum of $G_C = (C, E_C)$ having a high extinction value [33] and a high attribute \mathcal{A} .

When we want to perform a self-dual filtering, *i.e.*, to process at the same time both upper and lower level sets, we choose as tree \mathcal{T} the tree of shapes. Such an operator is a morphological shaping. In both Fig. 2 and Fig. 6, we use for attribute \mathcal{A} the circularity, $1 - l_{min}/l_{max}$, where l_{min} and l_{max} denote the respective lengths of the small-

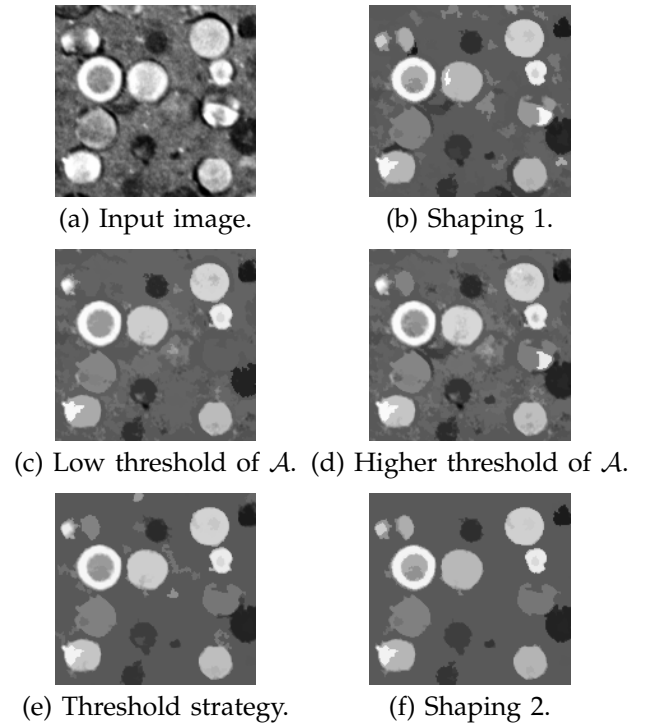


Fig. 6: Comparison between extinction-based shapings and attribute thresholding. (b-d): Using one shape attribute; (e-f): Using a combination of shape attributes.

est and of the largest axis of the best fitting ellipse of the node. The result of the shaping on Fig. 2 (a) is shown in Fig. 2 (g), and looks indeed better than the one of Fig. 2 (f). In Fig. 6, we compare a self-dual morphological shaping to a variant of the state-of-the-art thresholding approach [11]. In the latter, when the threshold of \mathcal{A} is low, some objects do not appear (Fig. 6 (c)). To be able to get all expected objects, we have to set a high threshold. However, in this case, too many unwanted objects are present (Fig. 6 (d)). The filtering strategy used in the shaping consists in preserving the minima of the graph G_C based on their extinction values [33]. More details about this filtering strategy can be found in [13]. With the shaping, all the expected objects can be found, as depicted in Fig. 6 (b). The results can be improved by combining some shape attributes. In Fig. 6 (e) and Fig. 6 (f), we use a combination of the circularity and of the non-compactness attribute, $\text{Tr}(I)/A^2$, where I is the moment-of-inertia tensor and A is the area [5, 11]. The combination of shape attributes significantly improves the results. Yet, the shaping in Fig. 6 (f) performs much better than the threshold-based strategy in Fig. 6 (e).

In Fig. 7, we give another example of comparison between our shaping and classical attribute thresholding. The used attribute function \mathcal{A} is the inverse of the context-based energy estimator proposed in [34]. Just like in active contours, this energy estimator measures the significance of each region, based on the image contents around the region boundary. Small values of the estimator correspond to meaningful objects. More details about it can be found in [34]. We construct a Min-tree

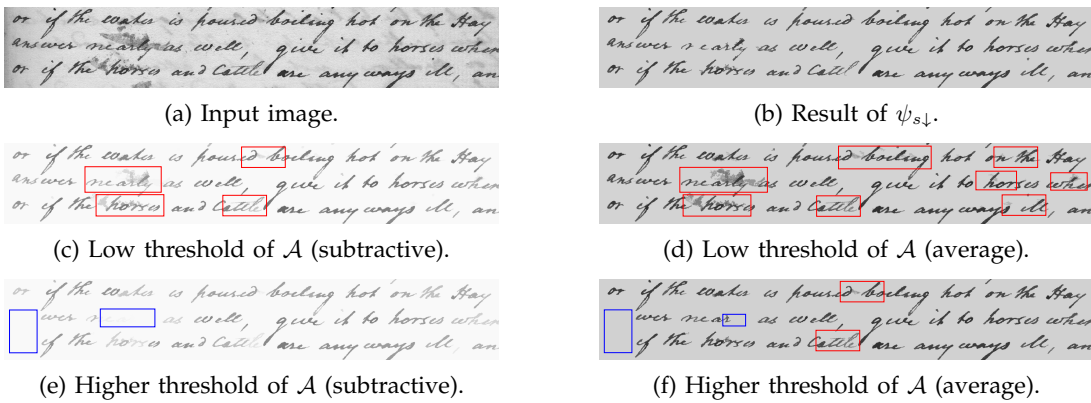


Fig. 5: Comparison of shape-based lower leveling to attribute thresholding; the attribute function is the average of the gradient magnitude on the shape contour.

\mathcal{TT} on G_C , and we choose for the second attribute \mathcal{AA} the total variation (as in Fig. 4). As depicted in Fig. 7, classical attribute thresholding gives reasonable results (in Fig. 7 (c) and (d)), but some unwanted structures in the background remain. In contrast, as observed in Fig. 7 (b), the shaping preserves all the expected structures on a clean background.

Shape-space filters provide more possibilities and flexibilities for image filtering as compared to classical attribute filters. In all those above illustrations, our shape-space filters give results that are more robust than the classical attribute thresholding methods. And some of them are impossible to obtain with classical methods.

6.2 Blood vessels segmentation in retinal images

In this section, we evaluate quantitatively one of the shape-space filters on the specific application of retinal blood vessel segmentation, where it is used as a segmentation tool. The motivation here is not to develop a complete process, but rather to demonstrate that the qualitative results of the previous section 6.1 are of some practical importance. In particular, we are going to see that a shape-based filter can advantageously replace its threshold-based counterpart.

As in many other existing approaches [35], we work here on the green channel of the color retinal image. To improve the visibility of the blood vessels, for each color retinal image f_c , a black top-hat transform, given by $\phi_{b_5}(f_g) - f_g$, where ϕ is morphological closing using a disk with radius 5 as structuring element b_5 , is applied to the green channel f_g . When a mask of eye fundus is available, we combine it with the black top-hat f_t . We thus obtain an image f_i in which the blood vessels are visible, and the main structures of the blood vessels are present in the Max-tree \mathcal{T} representation of f_i .

Blood vessels are usually long and thin structures. Hence, the attribute used here is the elongation \mathcal{A}_e . For each node C of the Max-tree \mathcal{T} , the elongation attribute is given by

$$\mathcal{A}_e(C) = |C|/(\pi \times l_{max}^2), \quad (14)$$

where $|\cdot|$ denotes the cardinality (area) and l_{max} denotes the length of the largest axis of the best fitting ellipse for

the connected component C . We expect that nodes with low attribute \mathcal{A}_e correspond to blood vessels.

The core of this application is the filtering of the Max-tree \mathcal{T} . A mere thresholding of the elongation \mathcal{A}_e already provides interesting results, but also gives unwanted objects (noise). However, a very low thresholding value t_{min} on \mathcal{A}_e ensures that thresholded nodes are blood vessels. These initially extracted nodes are used as seeds in the sequel. We then apply a morphological filtering with a depth criterion: using the Min-tree \mathcal{TT} of the node weighted graph $(G_{\mathcal{T}}, \mathcal{A}_e)$, we only preserve the nodes that have an height smaller than a given thresholding value d_0 in \mathcal{TT} and that furthermore contain the seeds. The connected components contained in the preserved nodes of \mathcal{TT} are considered as segmented blood vessels. The whole process is one of the many variants of shape-based upper levelings. An example of such a blood vessel segmentation process is given in Fig. 8. The elongation-based upper leveling correctly segments most of the blood vessels (see Fig. 8 (e)).

We have tested this specific shape-based upper leveling on the database of Digital Retinal Images for Vessel Extraction (DRIVE) [36, 37] and on the database of Structured Analysis of the Retina (STARE) [38, 39]. Fig. 9 shows two segmentation results on DRIVE database, where the parameter t_{min} is set to 0.05, and the parameter d_0 is set to 0.09. Two segmentation results on STARE database are depicted in Fig. 10 in Appendix A for this paper. Qualitatively, most of the blood vessels are correctly extracted, although some noise at the end of the vessels is also extracted. In addition some very thin blood vessels are missed.

Quantitative assessment is based on three performance measurements: *sensitivity*, *specificity*, and *accuracy* [37] (see Appendix A and [37] for details about these measurements). High values of these measurements are required for good segmentation results. A benchmark of our approach compared to the threshold-based attribute filtering [40], and one of the best dedicated method [35] on DRIVE database is provided in Table 1. The proposed elongation-based upper leveling is more robust than the thresholding-based subtractive filter given in [40], where the thresholding value is set manually for each

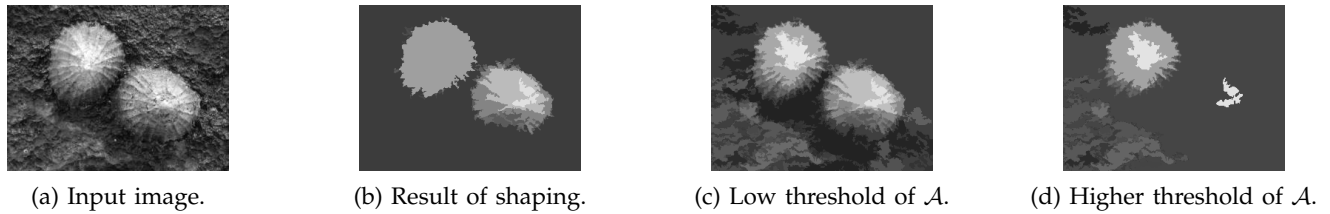


Fig. 7: Comparison between shaping and attribute thresholding, with as attribute the context-based energy estimator [34].

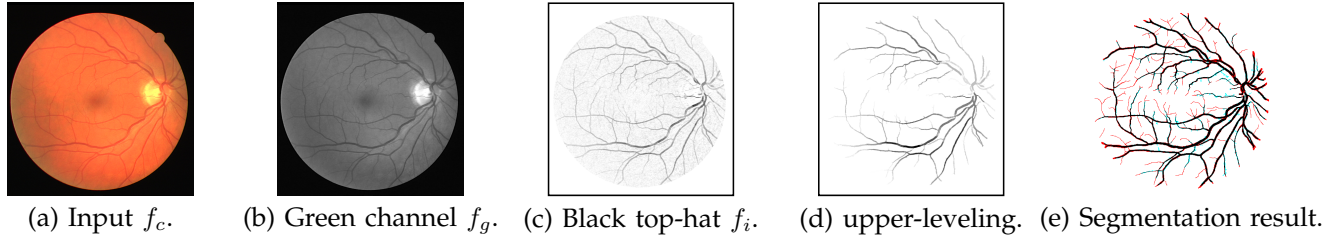


Fig. 8: Steps for the blood vessel segmentation in retinal image using elongation-based upper leveling. Images in (c) and (d) are reversed. In (e), black pixels are true positives, white pixels are true negatives, cyan pixels are false positives and red pixels are false negatives. t_{min} is set to 0.05, and d_0 is set to 0.09.

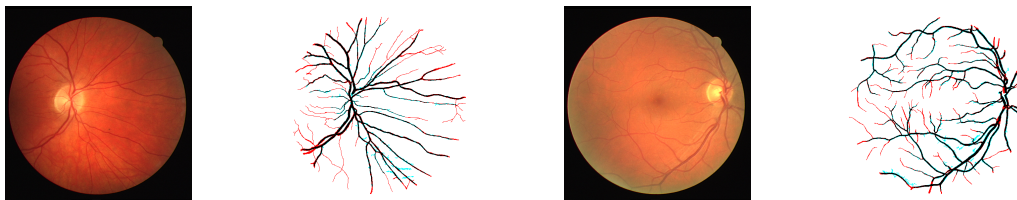


Fig. 9: Illustration of blood vessel segmentation on the DRIVE database. In the segmented results, black pixels are true positives, white pixels are true negatives, cyan pixels are false positives and red pixels are false negatives.

Method	DRIVE		
	<i>sensitivity</i>	<i>specificity</i>	<i>Accuracy</i>
2 nd Expert	0.7761	0.9725	0.9473 (0.0048)
Mendonça [35]	0.7344	0.9764	0.9452 (0.0062)
Our	0.6830	0.9788	0.9409 (0.0079)
Subtractive [40]	-	-	0.9389 (0.0056)

TABLE 1: Benchmark of several blood vessel segmentation approaches on DRIVE database.

image. In addition, in [40], a branch filtering [41] is also required to remove some unwanted neighboring objects of the expected objects or noise in the background. In contrast, the elongation-based upper-leveling achieves clean background while preserving expected objects. In the Appendix A of this paper, a complete benchmark of different methods on DRIVE and STARE databases is provided in Table 2 and in Table 3.

7 CONCLUSION

Connected operators are filtering tools that have very good contour preservation properties and are appropriate for both low level filtering and high level object recognition [4]. In this paper, we have expanded on the idea of tree-based connected operators. We propose to apply connected operators to an equivalent tree-based

representation of the image. We term such a processing shape-space filtering. We have shown that the framework encompasses the classical attribute filters. We have introduced two novel classes of connected filters: the shape-based lower/upper levelings (obtained with \mathcal{T} being the Min-tree or Max-tree) and the morphological shapings (obtained with \mathcal{T} being the tree of shapes). Those connected operators filter out image components based on some non-increasing criteria. Furthermore, we have studied some of the properties of those novel connected operators.

We have compared the shape-space filters to their corresponding threshold-based methods through some filtering illustrations. Shape-space filters usually provide more robust results (clean background while, at the same time, preserving the desired structures) and achieve some results that are impossible to obtain with classical approaches. We have evaluated one of the many shape-based upper levelings on retinal blood vessel segmentation, showing that it outperforms a dedicated approach relying on a classical threshold-based filter.

We believe that shape-space filtering has a large potential. In this paper, we have only presented the aspect of filtering, and hardly skimmed the other surface of what they have to offer to the scientific community. In [34], we have used this framework to perform object segmenta-

tion on the tree of shapes. The idea is to propose an efficient context-based energy estimator whose minima correspond to meaningful objects. Other criteria that can be used in this context are for example the number of false alarms [42] and the snake energy [43]. A practical problem is that many minima of such a criterion do not correspond to meaningful components of the image. In [34], a morphological closing in the space of shapes helps to filter out those spurious minima. When augmenting the filtering strength, more and more minima disappear. Indeed, the extinction value [33] measures the significance of minima, based on which we can produce extinction-based saliency maps [44, 45]. In [46], we have shown an example of such obtained saliency maps, which is an extension of the notion of constrained connectivity [47] to non-increasing constraints, using as an example a criterion from [48].

In future work, two different research directions are of interest. On the one hand, many other attributes than the ones tested in this paper need to be investigated. In this respect, *learning* the attributes rather than using hand-crafted ones is an interesting perspective. On the other hand, building G_C from other trees than the Min-tree, Max-tree, or the tree of shapes deserve further studies: obvious examples are the binary partition tree [49] (see [46] for some preliminary results), or tree-based shape spaces using mask-based second generation connectivity [7] or hyperconnectivity [8]. More generally, the proposed approach can further be generalized to other spaces, *i.e.*, an appealing idea is to build G_C from graphs that do not necessarily have a tree structure: family of segmentations, component-graphs of multivalued images [50, 51], directed graphs (considering non-symmetric adjacency relations [52, 53]) are promising examples of such graphs.

As a final note, we would like to stress out that the filters proposed in this paper are part of the C++ image processing library Milena [54], available on the Internet as free software. The present paper also demonstrates the interest of applying image processing algorithms to different types of graphs, in other words, the interest of genericity for image processing.

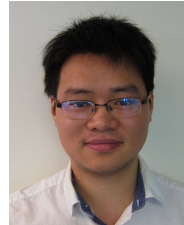
REFERENCES

- [1] J. Serra, *Image Analysis and Mathematical Morphology*. New York: Academic Press, 1982, vol. 1.
- [2] P. Salembier and J. Serra, "Flat zones filtering, connected operators and filters by reconstruction," *IEEE Transactions on Image Processing*, vol. 3, no. 8, pp. 1153–1160, 1995.
- [3] E. Breen and R. Jones, "Attribute openings, thinnings, and granulometries," *Computer Vision and Image Understanding*, vol. 64, no. 3, pp. 377–389, 1996.
- [4] P. Salembier and M.H.F. Wilkinson, "Connected operators," *IEEE Signal Processing Magazine*, vol. 26, no. 6, pp. 136–157, 2009.
- [5] M.A. Westenberg, J.B.T.M. Roerdink, and M.H.F. Wilkinson, "Volumetric attribute filtering and interactive visualization using the max-tree representation," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2943–2952, 2007.
- [6] M.H.F. Wilkinson, H. Gao, W.H. Hesselink, J.E. Jonker, and A. Meijster, "Concurrent computation of attribute filters on shared memory parallel machines," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1800–1813, 2008.
- [7] G.K. Ouzounis and M.H.F. Wilkinson, "Mask-based second generation connectivity and attribute filters," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 990–1004, 2007.
- [8] G.K. Ouzounis and M.H.F. Wilkinson, "Hyperconnected attribute filters based on k-flat zones," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 224–239, 2011.
- [9] P. Salembier, A. Oliveras, and L. Garrido, "Antiextensive connected operators for image and sequence processing," *IEEE Transactions on Image Processing*, vol. 7, no. 4, pp. 555–570, 1998.
- [10] P. Monasse and F. Guichard, "Fast computation of a contrast-invariant image representation," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 860–872, 2000.
- [11] E.R. Urbach, J.B.T.M. Roerdink, and M.H.F. Wilkinson, "Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 272–285, 2007.
- [12] Y. Xu, "Tree-based shape spaces: Definition and applications in image processing and computer vision," Ph.D. dissertation, Université Paris-Est, France, December 2013. [Online]. Available: <https://tel.archives-ouvertes.fr/tel-00965890/>
- [13] Y. Xu, T. Géraud, and L. Najman, "Morphological filtering in shape spaces: Applications using tree-based image representations," in *Proc. of Intl. Conf. on Pattern Rec.*, 2012, pp. 485–488.
- [14] F. Meyer, "From connected operators to levelings," in *Proceedings of the International Symposium on Mathematical Morphology*. Kluwer, June 1998, pp. 191–198.
- [15] F. Meyer, "Levelings, image simplification filters for segmentation," *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1-2, pp. 59–72, 2004.
- [16] J. Serra, "Connectivity on complete lattices," *Journal of Mathematical Imaging and Vision*, vol. 9, no. 3, pp. 231–251, 1998.
- [17] U. Braga-Neto and J. Goutsias, "A theoretical tour of connectivity in image processing and analysis," *Journal of Mathematical Imaging and Vision*, vol. 19, no. 1, pp. 5–31, 2003.
- [18] J. Serra and P. Salembier, "Connected operators and pyramids," in *Proceedings of SPIE International Symposium on Optics, Imaging, and Instrumentation*, 1993, pp. 65–76.
- [19] V. Caselles and P. Monasse, *Geometric Description of Images as Topographic Maps*, ser. Lecture Notes in Mathematics. Springer, 2009, vol. 1984.
- [20] L. Najman and T. Géraud, "Discrete set-valued continuity and interpolation," in *Proceedings of the International Symposium on Mathematical Morphology*, ser. Lecture Notes on Computer Science. Springer, 2013, vol. 7883, pp. 37–48.
- [21] H.J.A.M. Heijmans, "Connected morphological operators for binary images," *Computer Vision and Image Understanding*, vol. 73, no. 1, pp. 99–120, 1999.
- [22] P. Maragos, "A representation theory for morphological image and signal processing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, pp. 586–599, 1989.
- [23] J. J. Koenderink, "The structure of images," *Biological Cybernetics*, vol. 50, no. 5, pp. 363–370, 1984.
- [24] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proceedings of British Machine Vision Conference*, 2002, pp. 384–396.
- [25] M. Donoser and H. Bischof, "Efficient maximally stable extremal region (mser) tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006, pp. 553–560.
- [26] A. Desolneux, L. Moisan, and J. Morel, "Edge detection by helmholtz principle," *Journal of Mathematical Imaging and Vision*, vol. 14, no. 3, pp. 271–284, 2001.
- [27] Y. Xu, P. Monasse, T. Géraud, and L. Najman, "Tree-based morse regions: A topological approach to local feature detection," *IEEE Trans. on Image Processing*, vol. 23, no. 12, pp. 5612–5625, 2014.
- [28] O. Lezoray and L. Grady, *Image Processing and Analysis with Graphs: Theory and Practice*, ser. Digital Imaging and Computer Vision. Taylor & Francis, 2012.
- [29] L. Grady and J.R. Polimeni, *Discrete Calculus: Applied Analysis on Graphs for Computational Science*. Springer, 2010.
- [30] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, and p. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.
- [31] J. Crespo and R. W. Schafer, "Locality and adjacency stability constraints for morphological connected operators," *Journal of Mathematical Imaging and Vision*, vol. 7, no. 1, pp. 85–102, 1997.
- [32] E. R. Urbach, "Contextual image filtering," in *International Conference on Image and Vision Computing New Zealand*, 2009, pp. 299–303.
- [33] C. Vachier and F. Meyer, "Extinction values: A new measurement of persistence," *IEEE Workshop on Nonlinear Signal and Image Processing*, pp. 254–257, 1995.

- [34] Y. Xu, T. Géraud, and L. Najman, "Context-based energy estimator : Application to object segmentation on the tree of shapes," in *Proceedings of IEEE International Conference on Image Processing*, 2012, pp. 1577–1580.
- [35] A.M. Mendonça and A. Campilho, "Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction," *IEEE Transactions on Medical Imaging*, vol. 25, no. 9, pp. 1200–1213, 2006.
- [36] "Drive: Digital retinal images for vessel extraction," <http://www.isi.uu.nl/Research/Databases/DRIVE/>.
- [37] J. Staal, M.D. Abramoff, M. Niemeijer, M.A. Viergever, and B. van Ginneken, "Ridge based vessel segmentation in color images of the retina," *IEEE Transactions on Medical Imaging*, vol. 23, no. 4, pp. 501–509, 2004.
- [38] "Stare: Structured analysis of the retina," <http://www.ces.clemson.edu/~ahoover/stare/>.
- [39] A. Hoover, V. Kouznetsova, and M.H. Goldbaum, "Locating blood vessels in retinal images by piece-wise threshold probing of a matched filter response," *IEEE Transactions on Medical Imaging*, vol. 19, pp. 203–210, 2000.
- [40] I.K.E. Purnama, K. Aryanto, and M.H.F. Wilkinson, "Non-compactness attribute filtering to extract retinal blood vessels in fundus images," *International Journal of E-Health and Medical Communications*, vol. 1, no. 3, pp. 16–27, 2010.
- [41] I.K.E. Purnama, M.H.F. Wilkinson, A.G. Veldhuizen, P.M.A. van Ooijen, J. Lubbers, T.A. Sardjono, and G.J. Verkerke, "Branches filtering approach for max-tree." in *International Conference on Computer Vision Theory and Applications*, 2007, pp. 328–332.
- [42] F. Cao, P. Musé, and F. Sur, "Extracting meaningful curves from images," *Journal of Mathematical Imaging and Vision*, vol. 22, pp. 159–181, 2005.
- [43] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes : Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1987.
- [44] L. Najman and M. Schmitt, "Geodesic saliency of watershed contours and hierarchical segmentation," *IEEE Trans. on Pattern Analysis and Machine Intell.*, vol. 18, no. 12, pp. 1163–1173, 1996.
- [45] L. Najman, "On the equivalence between hierarchical segmentations and ultrametric watersheds," *Journal of Mathematical Imaging and Vision*, vol. 40, pp. 231–247, 2011.
- [46] Y. Xu, T. Géraud, and L. Najman, "Two applications of shape-based morphology: Blood vessels segmentation and a generalization of constrained connectivity," in *Proceedings of the International Symposium on Mathematical Morphology*. Uppsala: Springer Berlin Heidelberg, 2013, pp. 390–401.
- [47] 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.
- [48] P. Felzenszwalb and D.P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, sep 2004.
- [49] 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.
- [50] N. Passat and B. Naegel, "Component-trees and multivalued images: Structural properties," *Journal of Mathematical Imaging and Vision*, vol. 49, no. 1, pp. 37–50, 2014.
- [51] E. Grossiord, H. Talbot, N. Passat, M. Meignan, P. Tervé, and L. Najman, "Hierarchies and shape-space for PET image segmentation," in *Proceedings of the International Symposium on Biomedical Imaging (ISBI)*, April 2015, to appear.
- [52] B. Perret, J. Cousty, O. Tankyevych, H. Talbot, and N. Passat, "Directed connected operators: Asymmetric hierarchies for image filtering and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 6, pp. 1162–1176, June 2015.
- [53] B. Perret and C. Collet, "Connected image processing with multivariate attributes: an unsupervised markovian classification approach," *Computer Vision and Image Understanding*, vol. 133, pp. 1–17, April 2015.
- [54] 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 *Proceedings of IEEE International Conference on Image Processing*, 2010, pp. 1941–1944, <http://olena.lrde.epita.fr>.
- [55] M. Niemeijer, J. Staal, B. van Ginneken, M. Loog, and M.D. Abramoff, "Comparative study of retinal vessel segmentation methods on a new publicly available database," in *Proceedings of SPIE Medical Imaging*, vol. 5370, 2004, pp. 648–656.
- [56] F. Zana and J. Klein, "Segmentation of vessel-like patterns using

mathematical morphology and curvature evaluation," *IEEE Transactions on Image Processing*, vol. 10, no. 7, pp. 1010–1019, 2001.

- [57] B. Al-Diri and D. Steel, "An active contour model for segmenting and measuring retinal vessels," *IEEE Transactions on Medical Imaging*, vol. 28, no. 9, pp. 1488–1497, 2009.
- [58] X. Jiang and D. Mojon, "Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 1, pp. 131–137, 2003.
- [59] M. Marinez-Pérez, A. Hughes, A. Stanton, S. Thom, A. Bharath, and K. Parker, "Scale-space analysis for the characterisation of retinal blood vessels," in *Proceedings of Medical Image Computing and Computer Assisted Intervention*, 1999, pp. 90–97.



Yongchao Xu received in 2010 both the engineer degree in electronics & embedded systems at Polytech Paris Sud and the master degree in signal processing & image processing at Université Paris Sud, and the Ph.D. degree in image processing and mathematical morphology at Université Paris Est in 2013. He is currently working at EPITA Research and Development Laboratory (LRDE) and at the Signal and Image Processing Department, Telecom ParisTech as a postdoctoral fellow. His research interests include

connected filtering, image segmentation, medical image analysis, and local feature detection.



Thierry Géraud received a Ph.D. degree in signal and image processing from Télécom Paris-Tech 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.



Laurent Najman Laurent Najman received the Habilitation à Diriger les Recherches in 2006 from the University of Marne-la-Vallée, a Ph.D. of applied mathematics from Paris-Dauphine University in 1994 with the highest honour (Félicitations du Jury) and an Ingénieur degree from the Ecole des Mines de Paris in 1991. After earning his engineering degree, he worked in the Central Research Laboratories of Thomson-CSF for three years, working on some problems of infrared image segmentation using mathematical morphology. He then joined a start-up company named Animation

Science in 1995, as director of research and development. The technology of particle systems for computer graphics and scientific visualisation, developed by the company under his technical leadership received several awards, including the European Information Technology Prize 1997 awarded by the European Commission (Esprit programme) and by the European Council for Applied Science and Engineering and the Hottest Products of the Year 1996 awarded by the Computer Graphics World journal. In 1998, he joined OC Print Logic Technologies, as senior scientist. He worked there on various problem of image analysis dedicated to scanning and printing. After ten years of research work on image processing and computer graphics problems in several industrial companies, he joined the Informatics Department of ESIEE, Paris in 2002, where he is a professor and a member of the Laboratoire d'Informatique Gaspard Monge, Université Paris-Est Marne-la-Vallée. His current research interest is discrete mathematical morphology and discrete optimization.

APPENDIX A ADDITIONAL RESULTS FOR BLOOD VESSEL SEGMENTATION IN RETINAL IMAGES

Fig. 10 shows two segmentation results on the STARE database, where the parameter t_{min} is set to 0.055, and the parameter d_0 is set to 0.01.

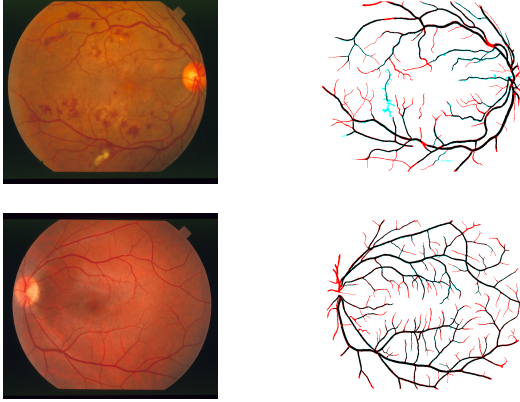


Fig. 10: Some blood vessel segmentation results on the STARE database. On the right side, black pixels are true positive, white pixels are true negative, cyan pixels are false positive and red pixels are false negative.

Quantitative assessment is based on three performance measurements: *sensitivity*, *specificity*, and *accuracy* [37]. Sensitivity measures the true positive rate (TPR), specificity measures the true negative rate (TNR), and accuracy measures the rate of pixels correctly classified. Another performance measurement, named *kappa* values, is also used in [37] for the DRIVE database. It is a statistical measure of inter-rater agreement, which measures the agreement between two raters who each classify N items into M mutually exclusive categories. These measurements are defined as: $sensitivity = TPR = TP/P = TP/(TP + FN)$, $specificity = TNR = TN/N = TN/(TN + FP)$, $accuracy = (TP + TN)/(TP + TN + FP + FN)$, $kappa = (Pb(a) - Pb(e))/(1 - Pb(e))$, where TP stands for true positive, FP for false positive, TN for true negative and FN for false negative. $Pb(a)$ is the relative observed agreement among raters, and $Pb(e)$ is the hypothetical probability of chance agreement. If the raters are in complete agreement then $kappa = 1$. If there is no agreement among the raters other than what would be expected by chance (as defined by $Pb(e)$), $kappa = 0$.

A complete benchmark of different methods is depicted in Table 2 and Table 3. In the case of DRIVE database (in Table 2), our result is slightly under the best results given by the method of Mendonça [35]. Note also that the approaches of Staal [37] and Niemeijer [55] are supervised approaches. On the STARE database (in Table 3), the proposed method performs also very well. It is better than the method of Mendonça [35]. Both methods are very close to the second human observer.

We would like to stress out that, as said in Section 6.2, our goal in this experiment was to demonstrate that

Method	DRIVE			
	TPR	TNR	Accuracy	Kappa
2 nd Expert	0.7761	0.9725	0.9473 (0.0048)	0.7589
Mendonça [35]	0.7344	0.9764	0.9452 (0.0062)	-
Staal [37]	0.7193	0.9773	0.9441 (0.0057)	0.7345
Niemeijer [55]	0.6793	0.9801	0.9416 (0.0065)	0.7145
Our	0.6830	0.9788	0.9409 (0.0079)	0.7126
Subtractive [40]	-	-	0.9389 (0.0056)	-
Zana [56]	0.6696	0.9769	0.9377 (0.0078)	0.6971
Al-Diri [57] -	-	-	0.9258 (0.0126)	0.6716
Jiang [58]	0.6478	0.9625	0.9222 (0.0070)	0.6399
Perez [59]	0.7086	0.9496	0.9181 (0.0240)	0.6389

TABLE 2: Benchmark of different blood vessel segmentation approaches on the DRIVE database. Staal’s [37] and Niemeijer’s [55] approaches are supervised.

Method	STARE		
	TPR	TNR	Accuracy
2 nd Expert	0.8949	0.9390	0.9354 (0.0171)
Jiang [58]	-	-	0.9513
Our	0.6955	0.9758	0.9460 (0.0113)
Mendonça [35]	0.6996	0.9730	0.9440 (0.0142)
Hoover [39]	0.6751	0.9567	0.9267 (0.0099)

TABLE 3: Benchmark of different blood vessel segmentation approaches on the STARE database.

shape-based filters can in many cases advantageously replace their threshold-based homologue. Hence, we did not try everything that is possible to do for blood vessel segmentation. Many other processing are possible, for example, it has been shown in [40] that using the three channels instead of only the green one can improve the results, as does image enhancement. Developing a complete application dedicated to the problem of retinal blood-vessel segmentation is left for further research.