Shape-based analysis on component-graphs for multivalued image processing

Eloïse Grossiord, Benoît Naegel, Hugues Talbot, Nicolas Passat, Laurent Najman

▶ **To cite this version:**

Eloïse Grossiord, Benoît Naegel, Hugues Talbot, Nicolas Passat, Laurent Najman. Shape-based analysis on component-graphs for multivalued image processing. 2018. hal-01695384v1

HAL Id: hal-01695384

https://hal.univ-reims.fr/hal-01695384v1

Preprint submitted on 29 Jan 2018 (v1), last revised 17 Sep 2018 (v3)

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.

Journal of Mathematical Imaging and Vision

Shape-Based Analysis on Component-Graphs for Multivalued Image Processing

--Manuscript Draft--

Manuscript Number:		
Full Title:	Shape-Based Analysis on Component-Graphs for Multivalued Image Processing	
Article Type:	Manuscript	
Keywords:	mathematical morphology; connected operators; morphological hierarchies; anti-extensive filtering; component-tree; component-graph; shaping; multivalued images.	
Corresponding Author:	Nicolas Passat Universite de Reims Champagne-Ardenne Reims CEDEX 2, FRANCE	
Corresponding Author Secondary Information:		
Corresponding Author's Institution:	Universite de Reims Champagne-Ardenne	
Corresponding Author's Secondary Institution:		
First Author:	Éloïse Grossiord	
First Author Secondary Information:		
Order of Authors:	Éloïse Grossiord	
	Benoît Naegel	
	Hugues Talbot	
	Nicolas Passat	
	Laurent Najman	
Order of Authors Secondary Information:		
Funding Information:	Agence Nationale de la Recherche (ANR-10-BLAN-0205)	Not applicable
	Agence Nationale de la Recherche (ANR-10-LABX-58)	Not applicable

Powered by Editorial Manager® and ProduXion Manager® from Aries Systems Corporation

Manuscript Click here to download Manuscript JMIV_EG.pdf

Click here to view linked References

Noname manuscript No.
(will be inserted by the editor)

Shape-Based Analysis on Component-Graphs for Multivalued Image Processing

Éloïse Grossiord · Benoît Naegel · Hugues Talbot · Nicolas Passat · Laurent Najman

the date of receipt and acceptance should be inserted later

Abstract Connected morphological operators based on hierarchical image models have been increasingly considered to provide efficient image segmentation and filtering tools in various application fields, e.g. (bio)medical imaging, astronomy or satellite imaging. Among hierarchical image models, component-trees represent the structure of grey-level images by considering their nested binary level-sets obtained from successive thresholds. Recently, a new notion of component-graph was introduced to extend the component-tree to model any grey-level or multivalued images. The notion of shaping was also recently introduced as a way to improve the anti-extensive filtering of grey-level images by considering a two-layer component-tree for grey-level image processing. In this article, we study how component-graphs (that extend the component-tree from a spectral point of view) and shapings (that extends the component-tree from a conceptual point of view) can be associated for the effective processing of multivalued images. We provide structural and algorithmic developments. The relevance and usefulness of such association are illustrated by applicative examples. This study opens the way to new paradigms for connected filtering based on hierarchies.

Keywords Mathematical morphology · connected operators · morphological hierarchies · anti-extensive filtering · component-tree · component-graph · shaping · multivalued images

Éloïse Grossiord (corresponding author)
LIGM, ESIEE, Université Paris-Est, CNRS, France, and Keosys, Nantes, France
E-mail: eloise.grossiord@esiee.fr
Tel: (+33) 1 45 92 66 15
Fax : (+33) 1 60 95 75 75

Benoît Naegel
ICube, Université de Strasbourg, CNRS, France
E-mail: b.naegel@unistra.fr

Hugues Talbot and Laurent Najman
LIGM, ESIEE, Université Paris-Est, CNRS, France
E-mail: {hugues.talbot,laurent.najman}@esiee.fr

Nicolas Passat
CReSTIC, Université de Reims Champagne-Ardenne, France
E-mail: nicolas.passat@univ-reims.fr

1 Introduction

Mathematical morphology is a well-known non-linear theory of image processing [1,2]. It was first defined on binary images, and then extended to the grey-level case [3]. Its extension to multivalued (e.g., colour, multispectral, label) images is an ongoing, important task, motivated by potential applications in multiple areas, such as medical imaging, remote sensing, astronomy or computer vision. Indeed, with the evolution of imaging technology, an increasing number of image modalities have become available: colour photography is an obvious example. In medical imaging, distinct image modalities are used in combination to provide complementary characteristics in the body; in remote sensing, sensors are used to generate a number of multispectral bands [4, 5]. There are many others.

In the framework of mathematical morphology, the basic algebraic structure is the complete lattice [6], namely a non-empty set of ordered elements, whose every non-empty subset admits an infimum and a supremum. This means that the definition of morphological operators requires the definition of an ordering relation between points to be processed. In the case of grey-level (resp. binary) images, the complete lattice is the partially ordered set of functions on $\mathbb{R}$ or $\mathbb{Z}$ (resp. on $\{0,1\}$), equipped with the point-wise partial ordering induced

from the canonical total order on the reals or integers. In the case for multivalued images, values are only canonically equipped with a partial ordering due to their vectorial nature. While this does not prevent the definition of a mathematically correct complete lattice structure, in applications this can create difficulties [7].

Several contributions have been devoted to this specific problem. A recent review can be found in [8]. Except in a small number of works (for instance in the case of label spaces [9,10]), most contributions intend to define relevant total orderings on multivalued spaces as originally described in [11]. In particular, two main ways were explored: splitting the value space into several totally ordered ones (marginal processing), or defining ad hoc total order relations [7], often guided by semantic considerations (vectorial processing), in particular for handling colour images [12,13,14] and less frequently multi- / hyperspectral images [15]: marginal ordering, conditional ordering (C-ordering, widely studied in the framework of colour morphology [7], including lexicographic ordering [16,14]), reduced ordering (R-ordering) [4], partial ordering (P-ordering), and more recently a combination of several of these orderings [12].

These approaches present the advantage of embedding multivalued images into models that simplify processing, similarly to grey-level images, in particular reducing algorithmic complexity. However, these simplifications of multivalued spaces potentially induce a loss of information intrinsically carried by these – more complex but richer – partially ordered value sets.

In this article – which is an improved version of the conference paper [17] – our purpose is to propose a new way to efficiently handle multivalued images in the framework of connected filtering. Our approach relies on the paradigm of connectivity, which models the spatial / structural links between some elementary patterns in an image. Intuitively (and informally), the notion of connectivity on a set Γ allows us to decide whether it is possible to move from a point (or a subset) of Γ to another, while always remaining in Γ. If this property is verified, we say that Γ is connected. Several (similar, and sometimes equivalent [18]) ways can be considered to define connectivity: from the standard notions of algebraic topology; from the notions of paths in digital / discrete spaces [19,20,21]; or by using morphological definitions of connectivity [6,22,23, 24,25].

In mathematical morphology, the notion of connectivity led to define many hierarchical data-structures, designed to model simultaneously the spatial and spectral information of an image. These data-structures induced various algorithmic approaches for image processing, lying in the family of connected operators. A brief state of the art on the notion of component-tree, which is the most popular morphological hierarchy data-structure, such as previous works about the hierarchical handling of multivalued images, are proposed in Section 2, in order to provide the context of our contribution. The formalism of component-tree and its extension to multivalued imaging, namely component-graphs, is then recalled in Section 3.

Our main contribution is proposed in Sections 4 and 5, where we describe how the notion of shaping, which consists of composing several component-trees at different semantic levels, can be extended to handle both component-trees and component-graphs, thus allowing the processing of multivalued images via a hierarchical approach. In particular, we provide in Section 4 a conceptual description of the extension of the classical anti-extensive filtering paradigm based on component-trees; while we provide a complete algorithmic description of the way to actually implement this filtering in the case of multivalued images taking their values in multiband spaces. Some application examples, provided in Section 6, illustrate the methodological interest of our hierarchical approach for handling multivalued images, via component-graphs and shapings.

2 Related works

2.1 Component-trees

The component-tree is a compact, information preserving, hierarchical representation of grey-level images. Induced by the inclusion relation between the binary components of successive level-sets obtained at different thresholdings, this structure models image characteristics in a mixed spectral-spatial space. By construction, the component-tree is well-adapted for the development of image processing and analysis methods, based on topology properties (connectivity), and aiming at extracting structures of interest with specific intensities (global / local extremal values).

From a methodological point of view, the efficiency of the component-tree relies on its low computation cost. Several efficient algorithms have been proposed for building the component-trees in quasi-linear time, in sequential [26,27,28] and in distributed ways [29]. A recent survey on the different computation algorithms can be found in [30]. This capacity to efficiently compute component-trees opens the way to the development of techniques, performed on an image via its component tree, that also present low algorithmic complexity.

As basic operations on tree nodes can be interpreted in terms of processing on the image, component-trees have been involved in several applications. Practically,

the proposed techniques have been designed to detect structures of interest using information computed on the nodes. In particular, filtering and segmentation [26, 31, 25] can be easily carried out by simply selecting nodes, leading to connected operators. The versatility of the component-tree structure has also led to many other applications, such as image retrieval [32], classification [33], interactive visualisation [34], or document binarisation [35].

The success of these methods relies on the development of efficient algorithmic processes for node selection. Two main approaches were developed to cope with filtering and segmentation issues. The first approach consists of minimizing an energy globally defined over the tree nodes, leading to define an optimal cut [36], that can be interpreted as a segmentation of the underlying image. These approaches are often formulated as optimization problems, where the space of solutions is composed of partitions from the hierarchy. This is the basis for carrying interactive segmentation [37]. Recently, the notion of braids of partitions [38] was introduced as a general framework for the optimization of segmentations based on hierarchical partitions. The second approach consists of determining locally the nodes that should be preserved or discarded, based on attribute values [39] stored at each node of the tree. The computed attributes are chosen according to the application context. This approach is formalized as an anti-extensive filtering framework [26,31], recalled in Section 4, and constitutes the methodological basis of the present work. The subtree obtained by pruning the component-tree of the image, with respect to these attributes, can then be used to reconstruct a binary (segmentation) or grey-level (filtering) result.

The main two limitations of the component-tree are (1) structural: it is heavily constrained by the topological structure of the image; and (2) spectral: it is limited to grey-level (i.e., totally ordered) value images. Structural extensions of the component-tree have been proposed in [40] to deal with ordered families of connectivities, leading to component-hypertrees, and in [41] to handle images defined as valued directed graphs, leading to directed acyclic graphs (DAGs) structured over a tree. Spectral extensions were first considered by exploring marginal approaches for colour image handling [42]. Then, actual extensions of component-trees to partially-ordered value images were pioneered in [43] and further formalized in [44]. Except in specific cases where the values are themselves hierarchically organized [45], the induced data-structure, namely a component-graph, is no longer a tree, but a DAG. The anti-extensive framework proposed for component-tree filtering remains valid in theory, but algorithmic issues have to be dealt with, both for node selection and image reconstruction [46,47].

2.2 Hierarchical approaches for multivalued image processing

Connected operators are effective image processing tools in the framework of mathematical morphology. They were intensively studied for the last twenty years [1,48, 26,49]. In this context, operators based on hierarchical image models (i.e., trees) have been the object of several structural, algorithmic and methodological developments [50], in order to tackle specific issues associated to various application fields.

When dealing with the extension of connected operators based on hierarchical image models[1] to multivalued image, two major approaches are generally considered: hierarchies of segmentations and morphological trees.

The first relies on hierarchical clustering. Indeed, hierarchical segmentation trees aim at either growing and merging regions in a bottom-up or splitting regions in a top-down fashion. (See [52] for a recent survey on hierarchical segmentations in the graph framework, as used in image processing.) The advantage of these methods when dealing with multivalued images is that they do not directly consider image levels but a simplifying metric during their construction process, i.e., they only rely on a distance function / norm between image values (e.g., a saliency measure for hierarchical watersheds [53], a merging order (region adjacency graph merging using techniques such as the irregular pyramid [54], constrained connectivity [55]) for partition trees (binary partition trees [56], α-trees [57], quadtrees [58]), or via hyperconnections [59]). The use of these intermediate functions "hides" the complexity of the space, but necessarily induces a bias in the constructed data-structure.

On the other hand, morphological trees focus on the inclusion relationship between components. These components result from thresholding operations on the image at all level. A total order on the image levels ensure the inclusion of the level sets. A total ordering is then required to compute this thresholding decomposition-based trees. Unfortunately, and contrary to grey-level images, the spaces in which multivalued images take their values are not canonically equipped with total orders. This is problematic as the hierarchy in the tree (such as component-tree-based processing) is driven by the total order of the image values. Then, building such

[1] The links that exist among most of the presented hierarchies are described in [51].

trees requires an ordering on intensities that has to be decided upon. Hence, morphological trees consist of simplifying the multivalued space of images *a priori* to retrieve tractable totally ordered values, e.g., by marginal or vectorial policies [5]. This strategy makes it possible to reuse morphological trees specifically designed for grey-level images, such as the component-tree [26] and its auto-dual version, the tree of shapes [60, 61]. However, this simplification of multivalued spaces induces a loss of information.

To cope with this problem, efforts have been conducted to extend these data-structures to more complex spaces. Nevertheless, preserving a tree structure still requires a final simplification [62], or restrictive constraints on the value space [45]. More specifically, for the extension of the tree of shapes to multivalued image, instead of imposing a total ordering on values, in [63], authors marginally compute the tree of shapes for each colour channel, and merge them into a single tree. The merging decision does not rely on values anymore but on properties computed in a shape space, potentially inducing a loss of coherence by merging unrelated shapes together. In [62], the inclusion relationship between the marginally computed tree of shapes enables the construction of a graph of shapes in a most efficient way. Since a natural total order on multivariate data is usually not obvious, some approaches try to deal with the inherent natural partial order. Consequently, a true extension of such hierarchies to multivalued spaces necessarily leads to a data-structures that is no longer a tree, but a directed acyclic graph. This is in particular the case for the notion of component-graph [44], that extends the component-tree. This approach uses directly the partial ordering of values and manipulates the underlying graph structure. The higher richness and structural complexity of the component-graph, with respect to the component-tree, induces algorithmic issues when considering the classical anti-extensive filtering process developed in [26,31]. This is in particular the case for handling spatial complexity [46], pruning policies and image reconstruction [47].

Recently, a new notion of shaping [64] was introduced as an efficient way to improve the framework of anti-extensive filtering of [26,31], by considering a two-layer component-tree for grey-level image processing [64,65]. In [17], we proposed to associate the notions of component-graphs and shaping for the effective processing of multivalued images, opening the way to new paradigms for connected filtering based on hierarchical representations.

3 Structural background notions

This article is set in the framework of vertex-valued graphs. We now recall some basic notions and notations on graphs. They will allow us to describe the component-tree and the component-graph in a simple and unified formalism, and to discuss, in Sections 4 and 5, how to carry out shaping on component-graphs to handle multivalued images. For the sake of clarity, Section 3 is written in a self-contained pedagogical way.

3.1 Order relations

Let Γ be a finite set of elements. Let $\leq$ be a (binary) relation on Γ. We say that $\leq$ is an order relation (and that $(\Gamma, \leq)$ is an ordered set) if $\forall x, y, z$, the relation $\leq$ satisfies the following conditions:

(i) $x \leq x$ (reflexivity);
(ii) $(x \leq y \wedge y \leq z) \Rightarrow (x \leq z)$ (transitivity); and
(iii) $(x \leq y \wedge y \leq x) \Rightarrow (x = y)$ (antisymmetry).

Moreover, we say that $\leq$ is a total (resp. partial) order relation (and that $(\Gamma, \leq)$ is a totally (resp. partially) ordered set), if $\leq$ is total (resp. partial) (i.e., if $\forall x, y \in \Gamma, (x \leq y \vee y \leq x)$ (resp. if $\exists x, y \in \Gamma, (x \not\leq y \wedge y \not\leq x)$)). The word "partial" indicates that there is no guarantee that two elements can always be compared

A partially ordered set can be modelled via its *Hasse diagram*, which depicts the ordering relation. More precisely, the Hasse diagram of an ordered set $(\Gamma, \leq)$ is its transitive reduction, i.e., the strictly ordered set $(\Gamma, \prec)$ such that for all $x, y \in \Gamma$, we have $x \prec y$ iff y *covers* x, i.e. $x < y$ and there is no $z \in \Gamma$ such that $x < z < y$. In the sequel, it is drawn so that elements are placed higher than (i.e., above) the elements they cover. The resulting diagram provides a compact and lossless description of the order relation $\leq$.

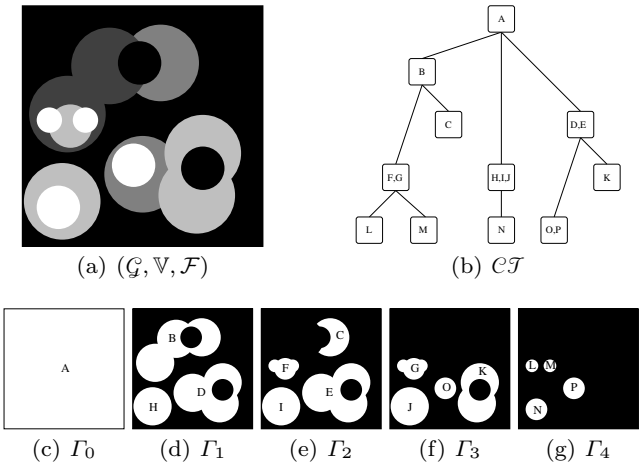
3.2 Vertex-valued graphs

We define a graph $\mathcal{G}$ as a couple $(\Gamma, \frown)$, where Γ is a nonempty finite set, and $\frown$ is a binary relation on Γ. The elements of Γ are called *vertices* or *points*. If two vertices x, y of Γ satisfy $x \frown y$, we say that they are *adjacent*; any such couple (x, y) is called an *edge*. A subgraph $\mathcal{G}'$ of $\mathcal{G}$ is a graph $(\Gamma', \frown)$ such that Γ' is a subset of Γ, equipped with the restriction of $\frown$ to Γ'.

In this work, we consider irreflexive graphs, i.e., we never have $x \frown x$. These irreflexive graphs are furthermore non-directed graphs, i.e., $x \frown y \Leftrightarrow y \frown x$; the edges (x, y) and (y, x) are then the same.

In $\mathcal{G}$, a *path* between two vertices x and y is defined as a sequence of distinct vertices of $\mathcal{G}$ from x to y such that any two successive vertices are adjacent. In this case, we say that x and y are connected in $\mathcal{G}$. If this path exists and is unique for any two vertices of the graph, then the graph is a *tree*. We say that $\mathcal{G}$ is *connected* if any two vertices of $\mathcal{G}$ are connected. The *connected components* of $\mathcal{G}$ are the maximal sets of vertices that can be linked by a path. The set of all these connected components is noted $\mathcal{C}[\mathcal{G}]$; it is a partition of Γ. Recall that a partition of Γ is a set $\mathcal{P}$ of nonempty disjoint subsets of Γ whose union is Γ i.e., $\forall X, Y \in \mathcal{P}, X \cap Y = \emptyset \; if X \neq Y and \; \cup \{X \in \mathcal{P}\} = \Gamma$.

Let $\mathcal{F} : \Gamma \to \mathbb{V}$ be a function such that $\mathbb{V}$ is a non empty finite set canonically equipped with an order relation $\leq$. The triple $(\mathcal{G}, \mathbb{V}, \mathcal{F})$ is called a *vertex-valued graph* (or *valued graph*, for brief). We now define the notions of component-tree and component-graph, based on this notion of valued graph.



(a) $(\mathcal{G}, \mathbb{V}, \mathcal{F})$ (b) $\mathcal{CT}$

(c) Γ_0 (d) Γ_1 (e) Γ_2 (f) Γ_3 (g) Γ_4

Fig. 1 (a) A grey-level image, viewed as a valued graph $(\mathcal{G}, \mathbb{V}, \mathcal{F})$, where $\mathbb{V} = [\![0, 4]\!] \subset \mathbb{Z}$ (from 0 in black; to 4 in white) equipped with the canonical order relation $\leq$. (c–g) Thresholded sets $\Gamma_v \subseteq \Gamma$ (in white) for v varying from 0 to 4. (b) The component-tree $\mathcal{CT}$ associated to $(\mathcal{G}, \mathbb{V}, \mathcal{F})$. The letters (A–P) in nodes correspond to the associated connected components (c–g).

3.3 Component-tree [26]

Let $(\mathcal{G}, \mathbb{V}, \mathcal{F})$ be a valued graph. In the sequel of this section, we assume that $\leq$ is a total order on $\mathbb{V}$, and that $\mathcal{G}$ is connected, i.e., $\mathcal{C}[\mathcal{G}] = \{\Gamma\}$ contains a unique connected component. Since Γ is finite, so is the set $\mathcal{F}(\Gamma) = \{\mathcal{F}(x) \mid x \in \Gamma\} \subseteq \mathbb{V}$. Without loss of generality, we can assume that $\mathbb{V} = \mathcal{F}(\Gamma)$ and is then finite. In particular, $(\mathbb{V}, \leq)$ admits a minimum, noted $\bot$.

For any $v \in \mathbb{V}$, we define the threshold set Γ_v by

$$\Gamma_v = \{x \in \Gamma \mid v \leq \mathcal{F}(x)\} \tag{1}$$

Any such threshold set Γ_v induces a subgraph $\mathcal{G}_v = (\Gamma_v, \frown)$ of $\mathcal{G}$. For any $v, v' \in \mathbb{V}$ we have $v \leq v' \Leftrightarrow \Gamma_{v'} \subseteq \Gamma_v$. In addition, for any connected component $X_{v'}$ of $\mathcal{C}[\mathcal{G}_{v'}]$, there exists a (unique) connected component X_v of $\mathcal{C}[\mathcal{G}_v]$ such that $X_{v'} \subseteq X_v$.

We note Ψ the set of all the connected components of the subgraphs $\mathcal{G}_v$ obtained by successive thresholdings of $\mathcal{G}$

$$\Psi = \bigcup_{v \in \mathbb{V}} \mathcal{C}[\mathcal{G}_v] \tag{2}$$

The component-tree of $(\mathcal{G}, \mathbb{V}, \mathcal{F})$, noted $\mathcal{CT}$, is the Hasse diagram of the partially ordered set $(\Psi, \subseteq)$. We can observe that $X \in \Psi$ can correspond to several connected components in distinct threshold sets $\Gamma_v \subseteq \Gamma$ for successive values $v \in \mathbb{V}$. Each $X \in \Psi$ is intrinsically associated in $\mathcal{CT}$ to a value $l(X)$, defined as the maximal value of $\mathbb{V}$ which generates this connected component by thresholding of $\mathcal{F}$.

As suggested by its denomination, the component-tree has a tree structure. Its vertices are also called nodes. Among them, the largest component is the maximum for the Hasse diagram, namely the set Γ, obtained as the unique connected component of $\mathcal{G} = \mathcal{G}_\bot$; it is the root of the tree.

On the opposite side, the leaves are the minimal elements of the Hasse diagram, i.e., the nodes of Ψ that do not strictly include any other nodes (see Figure 1). The order relation $\leq$ between nodes defines a parenthood relationship : a node n_1 is the *parent* of a node n_2 if $n_2 \subset n_1$ and if there is no other node n_3 such that $n_2 \subset n_3 \subset n_1$. In that case, we also say that n_2 is a *child* of n_1.

For image processing purposes, each node of $\mathcal{CT}$ generally stores a value: either an energy (for global optimization) or an attribute (for local selection); this value is most often real. In both cases, this valuation is modelled by a function $\mathcal{V} : \Psi \to \mathbb{R}$. In other words, such enriched component-tree can be itself interpreted as a valued graph $(\mathcal{CT}, \mathbb{R}, \mathcal{V})$.

3.4 Component-graph [44]

Let $(\mathcal{G}, \mathbb{V}, \mathcal{F})$ be a valued graph. We still assume that $\mathbb{V} = \mathcal{F}(\Gamma)$ is finite and that $(\mathbb{V}, \leq)$ admits a minimum, noted $\bot$. The graph $\mathcal{G}$ also remains connected, but the order relation $\leq$ on $\mathbb{V}$ need not be total.

We extend the notion of connected component in the following way. Let $X \in \mathcal{C}[\mathcal{G}_v]$ be a connected com-

ponent of the threshold set Γ_v inducing $\mathcal{G}_v$ at value v. (By opposition to totally ordered sets, there may exist several values $v_i \leq v$ ($i \in \mathbb{N}$) such that $X \subseteq X_i \in \mathcal{C}[\mathcal{G}_{v_i}]$ while v_i is a maximal value lower than v.) We define the couple $K = (X, v)$ as a *valued connected component*. We note Θ the set of all the valued connected components of $\mathcal{G}$, with respect to its successive thresholds, defined as

$$\Theta = \bigcup_{v \in \mathbb{V}} \mathcal{C}[\mathcal{G}_v] \times \{v\} \tag{3}$$

From the order relation $\leq$ defined on $\mathbb{V}$, and the inclusion relation $\subseteq$, we define[2] the order relation $\trianglelefteq$ on the valued connected components of Θ as

$$(X_1, v_1) \trianglelefteq (X_2, v_2) \Leftrightarrow \begin{cases} (X_1 \subset X_2) \vee \\ (X_1 = X_2 \wedge v_2 \leq v_1) \end{cases} \tag{4}$$

This order, which intuitively mixes the inclusion and the value orders between connected components in a lexicographic way, can be considered as an extension of the inclusion relation to valued connected components.

The component-graph, noted $\mathcal{CG}$, of the valued graph $(\mathcal{G}, \mathbb{V}, \mathcal{F})$ is the Hasse diagram of the partially ordered set $(\Theta, \trianglelefteq)$. It does not necessarily have a tree structure; indeed several paths may exist between two nodes. This derives from the fact that a node can be the child of several parents (and not only one, as in a tree). When a node is the child of two parent nodes that are not comparable in values, it means that either the two parent nodes meet without inclusion, i.e., they are not mutually included; or one of them is strictly included in the other.

The component-graph still has a unique "largest" node that is the maximum for the Hasse diagram, i.e., the node $(\Gamma, \bot)$; it is the root of the graph. Similarly to the component-tree, the component-graph still has leaves, that are the minimal elements of the Hasse diagram.

Three variants of the component-graph exist, by defining different subsets of valued connected components:

1. Θ (see Equation (3)) represents all the valued connected components induced by $\mathcal{G}$;
2. $\dot{\Theta}$ (see Equation (5)) corresponds to the set of the valued connected components of maximal values considering all connected components. The set of nodes $\dot{\Theta}$ provides at least one occurrence of a valued connected component for each possible support X induced by the image, while removing those that are hidden as the value v is lower; and
3. $\ddot{\Theta}$ (see Equation (6)) gathers the valued connected components which are generators of $\mathcal{F}$, i.e., those that actually contribute to the definition of the image support.

$$\dot{\Theta} = \{(X, v) \in \Theta \mid \forall (X, v') \in \Theta, v \not< v'\} \tag{5}$$

$$\ddot{\Theta} = \{(X, v) \in \Theta \mid \exists x \in X, v = \mathcal{F}(x)\} \tag{6}$$

Based on these definitions, we observe that

$$\ddot{\Theta} \subseteq \dot{\Theta} \subseteq \Theta \tag{7}$$

The Θ- (resp. $\dot{\Theta}$-, resp. $\ddot{\Theta}$-) component-graph of $\mathcal{F}$ is the Hasse diagram of the partially ordered set $(\Theta, \trianglelefteq)$ (resp. $(\dot{\Theta}, \trianglelefteq)$, resp. $(\ddot{\Theta}, \trianglelefteq)$).

The three variants of component-graphs present inverse relationships between computational cost and information richness: the representation is either highly informative but relies on expensive computational costs; or much lighter in terms of information richness but less costly. The set Θ, that models all the valued connected components in the image, is the most informative but also the most costly. The set $\dot{\Theta}$, which gathers the valued connected components with maximal level, is intermediate in terms of both cost and information. Finally the set $\ddot{\Theta}$, that is reduced to the minimal set of valued connected components needed to define the image support, is the least costly and informative. The relevance of each component-graph is directly dependent on the targeted image processing application.

An example of multivalued image $\mathcal{F}$ and its associated value set $\mathbb{V}$ are provided in Figure 2(a) and (b). Figure 2(c–l) depicts the various valued connected components obtained from this image. More precisely, the support X of a valued connected component is represented in white in each subfigure, while v is given by the value at which the image has been thresholded in the subfigure. The three variants of component-graphs are illustrated in Figure 2(m–o).

The component-graph is a relevant extension of the component-tree, as both (i) data structures are compliant for total ordered sets $(\mathbb{V}, \leq)$, hence compatible for grey-level images, and (ii) the component-graph satisfies the image (de)composition model associated to component-tree, defined later in the paper in Equation (10). In addition, similarly to component-trees, an attribute value can be stored in each node of the component-graph to characterize the corresponding component. The local node selection based on attributes can lead to filtering or segmentation strategies. This

[2] Practically, when $\leq$ is a total order, the component-graph and the component-tree are isomorphic [44]. Consequently, it would make sense to also consider the valued connected components and the order $\trianglelefteq$ for building the component-tree, as the threshold value that leads to the generation of a connected component is useful for image modelling and reconstruction; see Equation (10).

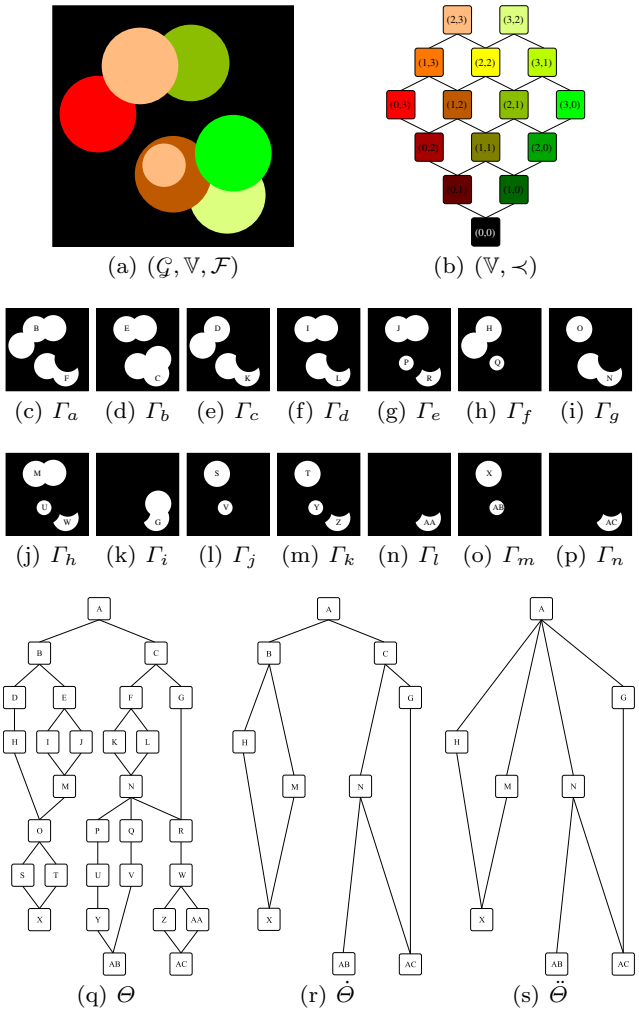


Fig. 2 (a) A multivalued image, viewed as a valued graph $(\mathcal{G}, \mathbb{V}, \mathcal{F})$, where $\mathcal{F} : \Gamma \to \mathbb{V}$ and $\mathbb{V} = \{(0,1), (1,0), (0,2), (1,1), (2,0), (0,3), (1,2), (2,1), (3,0), (1,3), (2,2), (3,1), (2,3), (3,2)\}$. (b) The Hasse diagram of the ordered set $(\mathbb{V}, \leqslant)$. For the sake of readability, each value of $\mathbb{V}$ is associated to an arbitrary colour. (c–p) Threshold sets Γ_v for $v \in \mathbb{V}$. (q–s) The Θ, $\dot{\Theta}$, $\ddot{\Theta}$-component-graphs of $\mathcal{F}$. The letters (A–AC) in nodes correspond to the associated connected components in (c–p). We can distinguish different types of valued connected components: they are either "completely visible" (e.g., the salmon/orange X), or "partially visible" (e.g., the green G), or "totally hidden" (e.g., the brown B). Those that are either *partially* or *totally visible* participate to the formation of the image support and then belong to $\Theta, \dot{\Theta}, \ddot{\Theta}$. Those that are *invisible* cannot belong to $\ddot{\Theta}$ but do belong to Θ. Among this set, the valued connected components that also belong to $\dot{\Theta}$ are those that present a maximal value v for a given support X.

valuation can also be interpreted as a function $\mathcal{A} : \Theta \to \mathbb{R}$. Then, such enriched component-graph is also interpreted as a valued graph $(\mathcal{CG}, \mathbb{R}, \mathcal{A})$.

4 Shape-space analysis of multivalued images

A (discrete) image $\mathcal{I}$ is a mapping from a finite spatial domain Ω (the image support, i.e., the set of its pixels / voxels) to a value space $\mathbb{V}$, equipped with an order relation $\leq$. For any $x \in \Omega$, $\mathcal{I}(x) \in \mathbb{V}$ is the value of $\mathcal{I}$ at x:

$$\left| \begin{array}{l} \mathcal{I} : \Omega \to \mathbb{V} \\ \quad x \mapsto \mathcal{I}(x) = v \end{array} \right. \tag{8}$$

Various choices are available for $\mathbb{V}$, such as $\mathbb{V} = \mathbb{R}^n$ or $\mathbb{V} = \mathbb{Z}^n$. The case of $n > 1$ corresponds to multivalued images, i.e., images with n components and $\mathbb{V} = \mathbb{V}_1 \times \ldots \times \mathbb{V}_n$, whose Cartesian product forms a complete lattice. Each mapping $\mathcal{I}_i : \Omega \to \mathbb{V}_i$ is called a band of the multivalued image; and v is a n-dimensional vector.

In order to develop morphological hierarchies, such as component-trees or component-graphs, it is required to know the order $\leq$ on $\mathbb{V}$. If $(\mathbb{V}, \leq)$ is a totally (resp. partially) ordered set, we say that $\mathcal{I}$ is a grey-level (resp. multivalued) image. For any $X \in \Omega$, and any $v \in \mathbb{V}$, we define the cylinder function $C_{(X,v)}$ of support X and value v, as:

$$\left| \begin{array}{l} C_{(X,v)} : \Omega \to \mathbb{V} \\ \qquad\qquad x \mapsto \begin{cases} v \text{ if } x \in X \\ \bot \text{ otherwise} \end{cases} \end{array} \right. \tag{9}$$

In addition, to develop connected operators, it is necessary to handle the structure of Ω, i.e., to know the adjacency between its points, leading to a graph $\mathcal{S}$. An image is then modelled as a valued graph $(\mathcal{S}, \mathbb{V}, \mathcal{I})$.

4.1 Anti-extensive filtering with the component-tree

The component-tree and the component-graph are image lossless models. Indeed, if we consider the image $\mathcal{I}$ in its functional form, i.e., as a mapping $\mathcal{I} : \Omega \to \mathbb{V}$, then $\mathcal{I}$ can be fully recovered from the (de)composition formula of Equation (10), i.e., the image can be expressed as the supremum of the nodes of its associated component-tree (Ψ) or component-graph (Θ):

$$\mathcal{I} = \bigvee_{(X,v) \in \Theta} C_{(X,v)} = \bigvee_{X \in \Psi} C_{(X, l(X))} \tag{10}$$

In the framework of component-trees (i.e., for grey-level images, i.e., when $\leq$ is a total order), this formula leads to a well-defined image for Ψ, but also for any subset of nodes $\widehat{\Psi} \subseteq \Psi$. In other words, it is possible to filter the image $\mathcal{I}$ by discarding some of the nodes of its hierarchical representation, and then reconstructing a resulting image from the preserved nodes. Each point $x \in \Omega$ in the filtered image then presents a value that is

lower or equal to the initial image; the induced operators are then anti-extensive. This anti-extensive filtering scheme was formalized for grey-level images in [26,31]. It basically consists of three successive steps:

(i) construction of the component-tree $\mathcal{CT}$ associated to the image $\mathcal{I}$;
(ii) reduction of the component-tree by selection of nodes $\widehat{\Psi} \subseteq \Psi$; and
(iii) reconstruction of the result image $\widehat{\mathcal{I}} \leq \mathcal{I}$ from the reduced component-tree $\widehat{\mathcal{CT}}$.

$$\begin{array}{ccc} (\mathcal{S}, \mathbb{V}, \mathcal{I}) & \xrightarrow{(i)} & (\mathcal{CT}, \mathbb{R}, \mathcal{V}) \\ \downarrow & & \downarrow (ii) \\ (\mathcal{S}, \mathbb{V}, \widehat{\mathcal{I}}) & \xleftarrow{(iii)} & (\widehat{\mathcal{CT}}, \mathbb{R}, \mathcal{V}_{|\widehat{\Psi}}) \end{array} \tag{11}$$

Step (i) is carried out from a wide range of available component-tree construction methods [30], while Step (iii) is straightforward from Equation (10). The restitution phase consists of transforming the output tree into an output filtered image. If a node has been removed, a new grey-level has to be assigned to it. Generally, this is the grey-level of the nearest preserved ancestor in the tree [26]. As a result, the grey-level values of the original image are assigned to the voxels of the preserved nodes, and no new grey-levels appear in the image.

The core of the process is Step (ii); it implies to choose a subset of nodes $\widehat{\Psi} \subseteq \Psi$. This choice is based on a selection criterion, i.e., a Boolean predicate that indicates if a node satisfies a required property; and a reduction policy to determine which parts of the component-tree should be kept or removed. More formally, considering attribute values carried by each node of the component-tree, namely the valuation $\mathcal{V} : \Psi \to \mathbb{R}$, guides the decision of preserving or discarding a node. If $\mathcal{V}$ models an increasing attribute, the restitution is simple because the tree branch is pruned, i.e., if a node is rejected, then all its descendants are also removed. However, if $\mathcal{V}$ models a non-increasing attribute, this phase is more difficult, as some rejected nodes can have preserved descendants. This means that the validity of the predicate for a given node does not imply its validity for the rest of the branch. Several classical policies have been defined, including in particular the Min, Direct and Max ones [26,31]:

- Min: a node is removed if it does not fulfill the criterion, or at least one of its parent node has been removed;
- Max: a node is removed if it does not fulfill the criterion, and all of its children nodes have been removed;
- Direct: a node is removed if it does not fulfill the criterion.

The Min/Max policies have a (sub)linear computational cost but might lack coherence in regards to the criterion. Indeed, they might discard/preserve nodes that meet/do not meet the criterion, depending on their position in the tree. The Direct policy only preserves the nodes fulfilling the criterion, but relies on an exhaustive scanning of all the nodes in the tree; its computational cost is then equal to the tree size.

4.2 Coupling shaping and component-graphs

In this section, we propose to describe how component-graphs (that extend the component-tree from a spectral point of view) and shaping (that extends the component-tree from a conceptual point of view) can be associated for the effective processing of multivalued images.

4.2.1 Extension of the anti-extensive filtering to component-graphs

The component-graph also satisfies the (de)composition formula classically associated to the component-tree (see Equation (10)). Indeed, an image $\mathcal{I}$ can be represented via the cylinder functions induced by the nodes Θ of its component-graph. In principle, we can then extend the above anti-extensive filtering to images taking their values in any value space $\mathbb{V}$, without the assumption that $\leq$ is a total order. We thus have to consider a component-graph, instead of a component-tree. This allows us to process any image in the same framework as initially proposed in [26,31]:

$$\begin{array}{ccc} (\mathcal{S}, \mathbb{V}, \mathcal{I}) & \xrightarrow{(i)} & (\mathcal{CG}, \mathbb{R}, \mathcal{A}) \\ \downarrow & & \downarrow (ii) \\ (\mathcal{S}, \mathbb{V}, \widehat{\mathcal{I}}) & \xleftarrow{(iii)} & (\widehat{\mathcal{CG}}, \mathbb{R}, \mathcal{A}_{|\widehat{\Theta}}) \end{array} \tag{12}$$

However, due to the complex structure of multivalued images and component-graphs, extending this framework is not straightforward and raises two difficulties. As the data-structure is no longer a tree, Step (ii) is now more complex. Indeed, even if pruning policies, defined for component-trees, remain consistent for component-graphs, they have to be adapted for dealing with non-linear bottom-up or top-down node parsing. Besides, Step (iii) can be an ill-posed problem, depending on the nature of the order $\leq$, and the preserved nodes $\widehat{\Theta}$. This issue is inherent to the component-graph structure. Indeed, if a node with several non-comparable

parents is removed during the filtering, the reconstruction can be challenging and subject to decisions (due to non-determinism). In addition, structural stability may be lost as the obtained image may correspond to a set of nodes that are different from those of the initial image.

4.2.2 Shaping: anti-extensive filtering in the shape-space [64]

The paradigm of shaping proposes to improve the framework of anti-extensive filtering to non-monotonic attributes, for grey-level image processing. It consists of performing the filtering on a double layer of component-trees, i.e., on the component-tree of the component-tree of the image, that transforms any attribute into a monotonic one.

Formally, the first layer corresponds to the component-tree $\mathcal{CT}$ of the image. Then, basically, the shaping consists of considering the component-tree $\mathcal{CT}$ itself as an image. Indeed, from a functional point of view, it can be defined as a mapping $\mathcal{CT} : \Psi \to \mathcal{V}$, where points are replaced by nodes, while intensities correspond to attribute values. Two nodes of the component-tree are adjacent if one of them if the parent of the other. This approach is tractable only if the space of the attribute values $\mathcal{V}$ is equipped with a total order relation, i.e., can be modelled as (a subset of) $\mathbb{R}$ or $\mathbb{Z}$; this is however the case for most attributes, in particular numerical ones. In this case, it is then possible to build a component-tree of this first tree $\mathcal{CT}$.

This "tree of tree" $\mathcal{CT}'$ can be processed as any other component-tree and we can perform anti-extensive filtering with it. It is then possible to process any grey-level image in the framework initially proposed in [26, 31], by performing node selection in a data-structure that is no longer defined at the image level, but at a higher semantic level. The virtue of this new tree is that the attribute computed from the nodes of $\mathcal{CT}$ is now increasing in $\mathcal{CT}'$; this allows us to perform real-time threshold-based node selection. The overall procedure remains quasi-linear in time and space, since we only duplicate the standard component-tree anti-extensive filtering process. The main limitation of this framework is that it considers a tree as intermediate data-structure, thus limiting its use to grey-level images.

$$\begin{array}{ccccc} (\mathcal{S},\mathbb{V},\mathcal{I}) & \xrightarrow{(i)} & (\mathcal{CT},\mathbb{R},\mathcal{V}) & \xrightarrow{(i')} & (\mathcal{CT}',\mathbb{R},\mathcal{V}) \\ \downarrow & & & & \downarrow{\scriptstyle (ii)} \\ (\mathcal{S},\mathbb{V},\widehat{\mathcal{I}}) & \xleftarrow{(iii')} & (\widehat{\mathcal{CT}},\mathbb{R},\mathcal{V}_{|\widehat{\Psi}}) & \xleftarrow{(iii)} & (\widehat{\mathcal{CT}}',\mathbb{R},\mathcal{V}_{|\widehat{\Psi}}) \end{array} \tag{13}$$

4.2.3 From "a tree on a tree" to "a tree on a graph"

The notion of valued graphs sheds light on the common structure of images, component-trees and component-graphs. In particular, it allows us to describe them with a simple and unified formalism. As a side effect, it emphasises the fact that shape-space filtering does not necessarily require a tree as inner layer, but it can also accept a graph. The cornerstone of this work is the generalization of the initial shaping paradigm: it can be used not only do build a "tree on a tree" but a "tree on a graph". This simple idea, summarized by Diagram (14), allows us – in theory – to process any image via a shape-based filtering.

$$\begin{array}{ccccc} (\mathcal{S},\mathbb{V},\mathcal{I}) & \xrightarrow{(i)} & (\mathcal{CG},\mathbb{R},\mathcal{A}) & \xrightarrow{(i')} & (\mathcal{CT},\mathbb{R},\mathcal{A}) \\ \downarrow & & & & \downarrow{\scriptstyle (ii)} \\ (\mathcal{S},\mathbb{V},\widehat{\mathcal{I}}) & \xleftarrow{(iii')} & (\widehat{\mathcal{CG}},\mathbb{R},\mathcal{A}_{|\widehat{\Theta}}) & \xleftarrow{(iii)} & (\widehat{\mathcal{CT}},\mathbb{R},\mathcal{A}_{|\widehat{\Psi}}) \end{array} \tag{14}$$

Based on the above remarks, this approach has the following virtues:

- it avoids the complex selection of nodes directly in the component-graph, since this task is indirectly carried out on the outer-layer component-tree;
- it extends the initial shaping approach beyond grey-level images to multivalued images;
- it inherits the good properties of shape-space filtering from increasing criteria, among which real time and interactive node selection at higher semantic level.

Nevertheless, behind this simple idea, and its intrinsic advantages, some algorithmic issues remain to be considered, in particular for the two reconstruction steps (iii'), (iii), from the component-tree to the component-graph, and then to the image. In Section 5, we propose some solutions to these issues.

5 Shape-space analysis of multivalued images: algorithmics

In this section, we now provide an algorithmic discussion about each step of the filtering framework depicted in Diagram (14), for handling multivalued images in the shape space. This algorithmic discussion is provided under the assumption that the considered multivalued images are multiband data, i.e., that the set of values $\mathbb{V}$ is composed of k spectral bands $\mathbb{V}_i$, each equipped with a

total order. In particular, we consider the canonical partial order $\leq$ on $\mathbb{V}$ defined by $(v_i)_{i=1}^k \leq (w_i)_{i=1}^k \Leftrightarrow \forall i \in [\![1,k]\!], v_i \leq w_i$. This hypothesis is motivated by the high frequency of such images in current image processing applications, which justifies to study them prioritarily.

5.1 Component-graph construction

Three variants of component-graphs were introduced in [44] (see Section 3.4), in particular to simplify $\mathcal{CG}$ by considering smaller subsets of Θ. In the first Step (i) of our framework, that builds the first layer component-graph $\mathcal{CG}$ from the initial multivalued image $(\mathcal{S}, \mathbb{V}, \mathcal{I})$, we chose to consider the lightest version $\ddot{\Theta}$ of component-graph (Figure 2(o)), i.e., the one that represents only the nodes which actually contribute to the construction of the image according to Equation (10).

Practically, this choice is motivated by several facts. First, we are considering multiband images, that generally correspond to "real" images, where values at each point have a physical meaning (by contrast, e.g., with semantic-content images, considered for instance in [45]). The $\ddot{\Theta}$-component-graph is the only one that guarantees that no new value will be introduced via a node initially "hidden" in the graph; this is a reasonable property in this context. For instance, it was observed in [46] that such component-graph was indeed relevant for filtering (i.e., denoising, simplification) purposes.

Second, from a space complexity point of view, this component-graph is much more efficient than the other two. Indeed, by construction, each node is "visible" in the modelled image; this means that at least one point of the image is directly and uniquely modelled by one node of the component-graph. A corollary of this property is that the number of nodes within $\ddot{\Theta}$ is bounded by the number of points of the image; in other words, the space complexity of the graph is (in the worst case) linear with respect to the image size. Since no intermediate superlinear data-structure is required for its construction, which may lead to extra computational cost, the building of this component-graph also presents a linear time complexity.

The component-graph $\mathcal{CG}$ is built from the algorithm recently proposed by some of the authors in [46]. For the sake of simplicity, we will now note $\ddot{\Theta}$ as Θ.

5.2 Component-graph valuation

At this stage, an attribute can be associated to each node of Θ, in the component-graph $\mathcal{CG}$, to retrieve a structure of valued graph, namely $(\mathcal{CG}, \mathbb{R}, \mathcal{A})$ in Diagram (14).

We chose to consider here an attribute taking its values in $\mathbb{R}$, namely a set where all values are comparable. While alternative choices are possible (see Section 7), we assume here that a valuation $\mathcal{A} : \Theta \to \mathbb{R}$ contains enough information to accurately filter the nodes, while leading to / forming a valued graph that authorises the building of a tree structure as second layer (that constitutes the very principle of shaping).

The criteria possibly modelled by $\mathcal{A}$ for each node $K = (X, v) \in \Theta$ can depend on:

(1) *spectral* properties, relying on image intensities, texture, etc. (i.e., the information stored in the v part, e.g., intensity mean, extrema, ...): then, we practically have $\mathcal{A} : \mathbb{V} \to \mathbb{R}$;
(2) *geometric* properties, relying on spatial information of the image (i.e., the information stored in the X part, e.g., compacity, flatness, ...): then, we practically have $\mathcal{A} : 2^\Omega \to \mathbb{R}$;
(3) *structural* properties, relying on the topology of the graph structure (e.g., branch length, position / rank of the node in the branch, ...): then, $\mathcal{A}(K)$ depends on the relationships of K and its neighbours within $\mathcal{CG}$;

or any combination of some of these three classes.

In contrast to the other two kinds of component-graphs, the chosen version of $\mathcal{CG}$ is relatively light. As a consequence, a criterion of type (3) would be weakly relevant, as all "internal" nodes of the graph are not modelled, thus making the graph structurally sparse. Indeed, the elimination of nodes from the richer variants of $\mathcal{CG}$ may hide information and introduce a bias in the graph structure, with respect to this specific type of attributes.

Then, only geometric criteria (type (2)) are considered here, for building the component-tree of the second layer. In particular, this choice is coherent with the paradigm of shaping – initially designed to focus on higher semantics at higher levels – and also motivated by the fact that the spectral handling (criteria of type (1)) of the image is intrinsically carried out at the first layer of the structure, since the component-graph already models the order between values in $\mathbb{V}$ via its organization; this spectral handling indeed happens both before (during the component-graph construction) or after the shaping stage (during the image reconstruction).

Another possibility is to define on this outer component-tree a second attribute $\mathcal{V}$ on which will be processed the tree pruning. In order to preserve the good properties of such filtering, it is essential that this new attribute $\mathcal{V}$ keeps the same behaviour as the first attribute $\mathcal{A}$, i.e., an increasing or decreasing evolution

along the tree. An example can be given by considering as attribute $\mathcal{V}$ the gap between the attribute value $\mathcal{A}_k$ of the node K and the value A_l of the leaves of its branch. Such criterion remains increasing, thus authorising a (relative) thresholding approach, similar to the first strategy but with a fine behaviour.

5.3 Component-tree construction and pruning

From the valued graph $(\mathcal{CG}, \mathbb{R}, \mathcal{A})$ associated to the component-graph, a shape-based component-tree can now be defined (Step (i') of our framework). This tree is the data-structure that will be considered for the pruning process (Step (ii)).

In practice, the structures of interest can be of two kinds, depending on the adopted processing paradigm: they are either structures to be preserved in the image (typically in a paradigm of segmentation), or structures to be removed (typically in paradigms of filtering or denoising). In both cases, the tree $\mathcal{CT}$ is built in order to model such structures as nodes located on the most distal parts of the tree, i.e., near the leaves.

Based on this fact, two basic policies can be considered to build the component-tree $\mathcal{CT}$, guided by the nature of the attribute $\mathcal{A}$, and more specifically by the correlation between the extremal values of $\mathcal{A}$ and the structures of interest. When these structures of interest correspond to the lowest values of $\mathcal{A}$, a min-tree representation (Figure 3(b)) is chosen, i.e., the root has the highest value, while the leaves have the lowest; when the structures of interest correspond to the highest values of $\mathcal{A}$, a – dual – max-tree representation (Figure 3(c)) is adopted. The inversion of the attribute allows to switch from one representation to the other.

Once the component – either min- or max- – tree has been built, the pruning process is carried out in a way that depends on the kind of processing paradigm. In the first case (segmentation paradigm), i.e., when structures of interest are to be preserved, relevant nodes are selected by preserving the distal parts, i.e., the branches of the tree (Figure 3(d)). In the second case (filtering paradigm), i.e., when structures of interest are to be eliminated, the pruning consists of eliminating those distal parts of the tree, i.e., preserving the proximal part (Figure 3(e)). In practice, the principle is to compute the most discriminative cut in the tree, and to preserve nodes located either below or above this cut, according to the chosen paradigm.

Practically, each node $Y \in \Psi$ of the component-tree $\mathcal{CT}$ is a connected component gathering nodes of a subgraph of $\mathcal{CG}$, for a given threshold value with respect to $\mathcal{A}$. This threshold value then constitutes the valuation

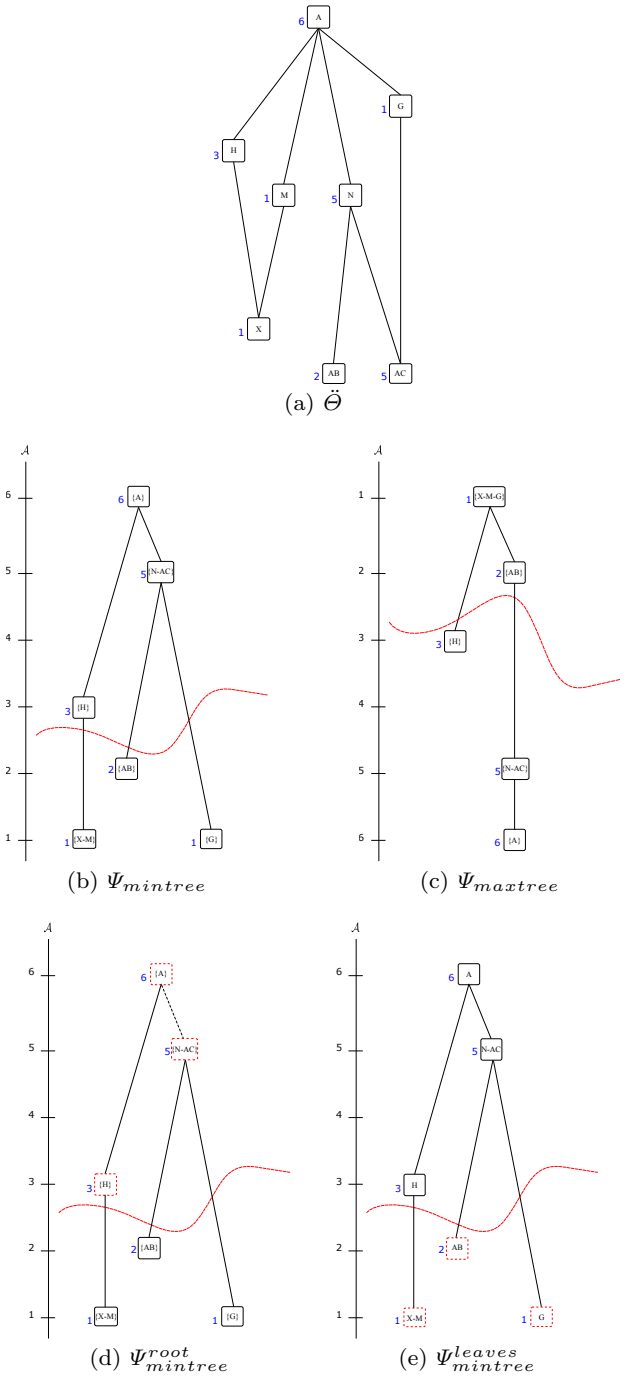


Fig. 3 Component-tree construction and pruning : (a) The $\ddot{\Theta}$-component-graph valued with attribute $\mathcal{A}$. (b) The min-tree $\Psi_{mintree}$ built from $\ddot{\Theta}$. (c) The max-tree $\Psi_{maxtree}$ built from $\ddot{\Theta}$. In these representations, a node of the component-tree may correspond to a node of the component-graph (in particular leaves nodes such as G in $\Psi_{mintree}$), or a set of nodes of the component-graph due to the inclusion relationship (for example, {A} gathers all the nodes of {A,G,H,M,N,X,AB,AC} in a same node in $\Psi_{mintree}$. The red line represents the cut in the tree. Depending on the policy, the nodes to be preserved are either located below the cut (c) or above it (d).

of this node. Following the above criteria classification, this new valuation $\mathcal{A}$ – that is however directly obtained from a valuation of class (2) – is now a valuation of class (1) in the shape-space. In addition, it defines a monotonic criterion, allowing for an easy selection by thresholding, and avoiding the use of any specific (e.g., Min, Max) pruning policies.

5.4 Component-graph filtering

A "standard" component-tree – defined from a grey-level image – contains nodes which represent connected components of points of the image, obtained at a given threshold value. By contrast, the component-tree $\mathcal{CT}$, defined at the outer layer of the shape-space model – computed from the valued graph $(\mathcal{CG}, \mathbb{R}, \mathcal{A})$ – contains nodes that are connected components of Θ, which are themselves connected components of Ω. Such nodes $Y \in \Psi$ are thus defined as sets $\{K_i = (X_i, v_i)\}_{i=1}^{k} \subseteq \Theta$, with $k \geq 1$.

Once the pruning process, carried out on the component-tree, has reduced the number of nodes Y preserved in $\widehat{\Psi}$, several policies can be considered to determine which nodes $K_i \in Y$ have to be preserved in the resulting "pruned" component-graph $\widehat{\mathcal{CG}}$, i.e., which nodes should form $\widehat{\Theta}$ (Step (iii)).

The easiest and most intuitive choice is to keep all the K_i contributing to the preserved Y, since they all provide spectral information in the image. However, since spectral information is already taken into consideration within the first layer of the structure, via the component-graph, while the geometric attributes rely on spatial information, it is reasonable to exclusively consider the spatial information handled by the preserved nodes Y. See Figure 4.

In particular, each node $K_i \in Y$ is either included in another node $K_j \in Y$, or is a maximal element in Y with respect to the $\trianglelefteq$ relation. When dealing with geometric criteria, only these latter nodes, that contribute to define the support $\bigcup_{i=1}^{k} X_i$ of Y in Ω, are of actual interest. In other words, if Y is preserved in $\widehat{\Psi}$, only these nodes may be preserved (both spatially and spectrally) in the filtered image. We note $\widehat{Y} \subseteq Y$ the subset of Y formed by such nodes.

The other nodes of Y are not taken into account and are lost in the representation. Practically, this is not a problem in general; indeed, most[3] of nodes $K \in \Theta$ belong to $\widehat{Y}$ for at least one $Y \in \Psi$. In other words, even by preserving a strict part of the nodes within the elected Y, each node K still has a chance to be finally preserved, thus avoiding erroneous removals.

The main difference between the initially proposed shaping paradigm ("a tree on a tree") and the present one ("a tree on a graph") is that the first defines any $\widehat{Y}$ as a singleton set $\{K\}$, while the second can now associate several – overlapping – nodes of Θ into a same $\widehat{Y}$, since some values of $\mathbb{V}$ may be non-comparable.

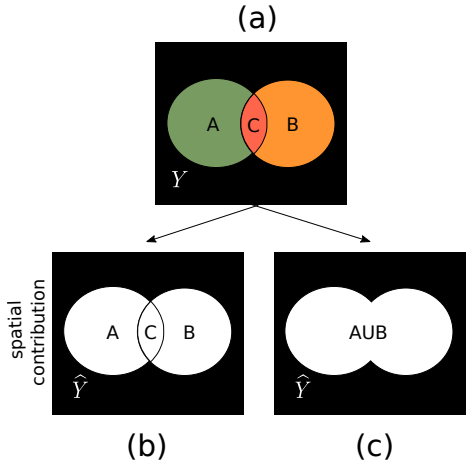


Fig. 4 Component-graph filtering: (a) Y, node from the component-tree preserved in $\widehat{\Psi}$ during pruning. Y is composed of three nodes of the component-graph $\mathcal{CG}$: A, B and C such that C $\in$ A and C $\in$ B. Two choices can be considered for the filtering of Y: (b) either all nodes (A, B and C) are preserved in $\widehat{Y}$ as they all contribute to the spectral values of the component; or (c) only A and B are kept in $\widehat{Y}$ when C is eliminated since $A \cup B$ define the spatial support of Y while C does not participate to the definition of the boundaries of Y; then the reconstructed intensities are determined thanks to policies described in Section 5.5.

5.5 Image filtering

The data-structure considered for processing an image is composed here of two layers of component-graph / tree. As a consequence, the final filtering of the image has to go successively through these two layers, thus leading to two filtering steps.

(1) The first step is a temporary reconstruction at the component-tree level: it consists of reconstructing regions of the image corresponding to each reduced node $\widehat{Y}$, associated to each node $Y \in \widehat{\Psi}$, based on the set of nodes $K_i = (X_i, v_i) \in \widehat{Y}$.

[3] Remark: In [17, p. 453], it was written that "*any node $K \in \Theta$ belongs to $\widehat{Y}$ for at least one $Y \in \Psi$*". This is not verified in all cases; a trivial counter-example is the case where Θ has at least 2 nodes, while the same value is given by $\mathcal{A}$ to each $K \in \Theta$, thus leading to a degenerated component-tree $\mathcal{CT}$ composed by a single node. However, for "real" valuation functions $\mathcal{A}$, and in particular those taking their values in $\mathbb{R}$, it is quite probable that most nodes $K \in \Theta$ belong to $\widehat{Y}$ for at least one $Y \in \Psi$.

(2) A given node $K \in \Theta$ may belong to $\widehat{Y_j}$, for several nodes $Y_j \in \widehat{\Psi}$, in that case, the second step – which leads to the final reconstruction of the image – handles the conflicting intersection between those reconstructed regions.

We recall that we deal here with multivalued images, and we assume that the value space $(\mathbb{V}, \leq)$ is structured as a lattice. As a consequence, the value set is an ensemble of ordered elements which admits an infimum and a supremum; this assumption is used hereafter for reconstruction purpose.

For the first step, various approaches can be considered to reconstruct image regions associated to the nodes $\widehat{Y}$ of the reduced component-tree. The first, and most straightforward possibility, is to preserve all values v_i associated to each node K_i of the reduced node $\widehat{Y}$; and delegate the real image valuation to the second step. We do not describe how to do that in the sequel of the paper. Instead, we focus on the second possibility that consists of assigning a unique value v – that can be defined with different policies – to the whole reduced node $\widehat{Y}$, thus creating a flat zone of value v, over the whole set of nodes $(K_i, v_i) \in \widehat{Y}$. In this case, we mainly have two options for defining v:

- v can be set as the *supremum* of all the v_i of each node $K_i = (X_i, v_i) \in \widehat{Y}$. However, this policy is not optimal as it leads to the loss of the anti-extensivity property of the designed filters. Indeed, the choice of the supremum creates connected components, possibly presenting a high value that did not exist in the initial set of connected components of the original image; or
- v can be defined as the *infimum* of all the v_i of each node $K_i \in \widehat{Y}$. This policy is more reasonable, as it ensures to preserve the property of anti-extensivity of the subsequent filters. This strategy is justified by the fact that the node Y has been preserved with respect to a geometrical attribute, computed for the union of all supports X_i of the K_i; in such conditions, the least common threshold value associated to all these nodes should be considered. Nevertheless, this strategy tends to "flatten" the intensities in the image and create connected components valued with intensities that are lower or equal to that of the existing support.

The second step consists of handling the conflicts of intensities assigned to nodes K belonging to several nodes $Y_j \in \widehat{\Psi}$. As in the first step, two policies can be considered.

(i) In the case where v was defined by a *supremum* paradigm:

- the value finally assigned to the node K can be defined as the *supremum* of all the conflicted values. This may tend to create connected components valued with higher or maximal intensities, that did not exist in the original image. The reconstruction of the filtered image can then be formalized as follows:

$$\widehat{\mathcal{I}} = \bigvee_{Y \in \widehat{\Psi}} C_{(\bigcup_{(X,v)\in\widehat{Y}} X, \bigvee_{(X,v)\in\widehat{Y}} v)} \tag{15}$$

- alternatively, the value assigned to the node K can be defined as the *infimum* of all the values in conflict. This policy will only lower the intensity of the connected components created at step one, or enlarge their spatial support. The spectral or spatial properties of the image are possibly lost as the resulting components did not exist in the initial image. The reconstruction of the filtered image can then be formalized as follows:

$$\widehat{\mathcal{I}} = \bigwedge_{Y \in \widehat{\Psi}} C_{(\bigcup_{(X,v)\in\widehat{Y}} X, \bigvee_{(X,v)\in\widehat{Y}} v)} \tag{16}$$

(ii) In the case where v was defined by an *infimum* paradigm:

- the value finally assigned to the node K can be defined as the *supremum* of all these values in conflict. This policy is justified by the fact that a node $Y \in \Psi$, defined as the union of several nodes of Θ, should not lose its geometry in the filtered image. The choice of the supremum enables the construction of the cylinder function $C_{(X,v)}$ of support X and value v (as defined in Equation (9)). It allows us to offset the flattening of intensities (due to the infimum policy at step one) and to come up with intensities closer to the ones of the original support. The reconstruction of the filtered image can then be formalized as follows:

$$\widehat{\mathcal{I}} = \bigvee_{Y \in \widehat{\Psi}} C_{(\bigcup_{(X,v)\in\widehat{Y}} X, \bigwedge_{(X,v)\in\widehat{Y}} v)} \tag{17}$$

- alternatively, the value assigned to the node K can be defined as the *infimum* of all values in conflict. This policy is nevertheless not reasonable because it will tend to completely flatten the intensity in the image; and the resulting support may be spectrally far from the real intensities. Besides, the choice of the infimum may result in the loss of the geometry of individual nodes Y due to the building of large flat components. The reconstruction of the filtered image can then be formalized as follows:

$$\widehat{\mathcal{I}} = \bigwedge_{Y \in \widehat{\Psi}} C_{(\bigcup_{(X,v)\in\widehat{Y}} X, \bigwedge_{(X,v)\in\widehat{Y}} v)} \tag{18}$$

One may notice that these various strategies, although presented in the general case of handling all bands of a multivalued image, can be restricted to a strict subset of bands. In the case where only one band is considered, the reconstructed image is a grey-level one, and the supremum and infimum on $\mathbb{V}$ considered above are simply replaced by the maximum and minimum in the considered band.

6 Application examples

In this section, we illustrate the relevance of the proposed framework on two application examples. All the presented results are based on the $\ddot{\Theta}$ component-graph.

The first application (Section 6.1) is the filtering of a real colour image. In this context, the multiband image corresponds to a function taking its values in a standard space (e.g., RGB, HSV), and the multivalued set $\mathbb{V}$ is equal to $[\![0, 255]\!]^k$, namely a space with several values in each band, but few bands ($k \leq 3$). This application consists of the filtering of a natural scene, by considering the issue of noise reduction.

The second application (Section 6.2) is the segmentation of multi-modality images. In this context, the multiband image corresponds to a function taking its values in a space with many values in each band, and two bands corresponding each to a given imaging modality, namely morphological X-ray Computed Tomography and functional Positron Emission Tomography. The segmentation of such images is designed to emphasise tumours based on their shape and metabolic activity.

The purpose of these examples is to qualitatively emphasise that the proposed framework is versatile enough for allowing us to process a wide range of useful applications. In particular, since this framework is very general, our purpose is not to prove that our approach is more efficient than others, but that it opens the way to alternative possibilities of image processing of multivalued images via connected operators. A complete application relying on this framework is a topic for further research.

6.1 Colour image filtering

6.1.1 Colour space

The processing of natural images, and in particular colour images, constitutes a major application field of multivalued mathematical morphology. As stated in Section 1, most approaches consist of defining ad hoc total orders on standard colour spaces, e.g., RGB or HSV value spaces, or splitting these orders into total orders on each band, with the purpose to take advantage of the existing tools initially designed for grey-level image processing, in particular in the framework of connected operators.

In this first experiment, we aim to investigate how the proposed framework can be used for processing colour images, by directly considering the standard orders on colour spaces. In this context, working with the chromatic bands is not relevant, since we do not make specific assumptions on objects sought in images. As a consequence, the HSV space was considered, instead of the RGB space. In addition, we only focus on the SV part of this space, as the H part carries – as RGB – a chromatic information that is not prone to be modelled by a natural order.

Let us consider a colour image $\mathcal{I} : \Omega \to \mathbb{V}$, where $\mathbb{V} = H \times S \times V$ is the HSV space. Each of the bands is an interval of integers; we only focus on the last two, namely S and V. As a consequence, we will let untouched the chromatic band H, and we can then only consider the image as a bi-band function $\mathcal{I} : \Omega \to [\![0, N-1]\!]^2$ (in our application, we will set $N = 256$). For each point x of Ω, we have $\mathcal{I}_i(x) = (s, v)$; $s = 0$ (resp. $N-1$) means that the colour at x is completely unsaturated, i.e., "grey" (resp. completely saturated); while $v = 0$ (resp. $N-1$) means that the colour at x has the darkest value, i.e., black (resp. the brightest value).

The set of values $S \times V = [\![0, N-1]\!]^2$ is composed of couples, and then has a cardinal of N^2. According to the nature of the structures that we wish to emphasise in the image, or the "noise" we wish to lower, the choice of the order $\leq$ on $S \times V$ differs. Indeed, we can define 4 orders, by defining $(s_1, v_1) \leq (s_2, v_2)$ by:

$$(s_1 \leq s_2) \wedge (v_1 \leq v_2) \tag{19}$$

$$(s_1 \leq s_2) \wedge (v_1 \geq v_2) \tag{20}$$

$$(s_1 \geq s_2) \wedge (v_1 \leq v_2) \tag{21}$$

$$(s_1 \geq s_2) \wedge (v_1 \geq v_2) \tag{22}$$

All these orders are partial, and lead to a structuration of $S \times V = [\![0, N-1]\!]^2$ as a complete lattice, organized into $N + 1$ layers.

We propose to filter some natural images $\mathcal{I} : \Omega \to [\![0, N-1]\!]^2$ in the HSV space, modelled as valued graphs $(\mathcal{S}, [\![0, N-1]\!]^2, \mathcal{I})$, where $\mathcal{S}$ is the graph corresponding to the topological structure of Ω. In other words, we only focus on the saturation and brightness of these images, while letting untouched the hue.

6.1.2 Salt-and-pepper removal

We have considered natural images corrupted with salt-and-pepper impulse noise, with various noise densities

Fig. 5 Filtering results (second row) for different noise rate using attribute-based component-graph : 10% (First column), 30 % (second column), and 50% (third column).

(varying from 1% to 90%). The salt-and-pepper noise was generated independently on each color (R, G and B) channels so the produced noise may be more natural, before converting into the HSV space. In this space, as explained before, the computing of the component-graph relies on the S and V components, while the original H component is restored in the filtered image.

For images corrupted with salt-and-pepper noise, noisy pixels can only take the maximum (*salt* noise) and minimum (*pepper* noise) values in the dynamic range. Based on that specific noise structure and on the filter spectral properties, our denoising scheme will consist of two similar consecutive steps, each enabling the elimination of one type of noise values.

Due to the conversion from RGB to HSV, noisy pixels defined on the RGB image will be in spatial correspondence in the S and V bands. The order between those channels can then be formalized by defining $(s_1, v_1) \leq (s_2, v_2)$ such that $(s_1 \leq s_2) \ \wedge \ (v_1 \leq v_2)$, in order to represent the most saturated and bright structures as maximal values in the graph. As salt-and-pepper noise can be characterized as randomly distributed dots of one or more pixels (speckles), in that representation, the criterion considered for filtering is the compactness factor [65], defined as the ratio between the extremal eigenvalues of the matrix of inertia. The filtered image is reconstructed in $\mathbb{N}^2$, following the policy proposed in Equation (17).

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

Fig. 7 The two consecutive steps of the salt-and-pepper filtering process, based on a compactness-based filtering on component-graph. (a) Original lena image. (b) Initial lena image corrupted with a salt-and-pepper noise with a density of 10 %. (c) Intermediate result of salt noise removal. (d) Final image with pepper noise filtering.

(a) (b) (c)

Fig. 8 Filtering results for recursive filtering steps (with different attributes) on lena image affected with salt-and-pepper noise of 50 %. First column: original noisy image. Second column: intermediate denoised image after elimination of compact noise. Third column: final image after removal of compact and elongated noisy structures.

The experimental PSNR results (Figure (6)) show good denoising power of the filter (Figure 5). Indeed, the first step (Figure 7(c)), which consists of the *salt* noise removal, enables to eliminate the small bright textural parts of the image, while leaving most of the other parts unchanged. On the other side, the second step (Figure 7(d)) i.e. *pepper* noise filtering, efficiently eliminates the dark noisy speckles. However, some details of the original image can be smeared by the filter when they correspond to the considered feature. We observe that phenomenon in the textured leaves of lena's hat or near her right eyes on Figure 7(d). Furthermore, we can note that our filter can remove noise of different size and shape, while maintaining the sharpness of structures. Indeed, when the noise rate is over 30 %, some noisy pixels can randomly constitute elongated structures. In such cases, the first filtering process, aiming at removing small compact structures will not be enough to eliminate all noisy structures, and the most noisy elongated objects will remain in the filtered image (Figure 8(b)). One simple way to improve this limitation is to employ a second filtering process (Figure 8(c)), to remove those remaining structures. We observe on the PSNR curve (Figure (6)) that the combination of two-passes filtering, corresponding to the blue curve, outperform the one-pass filtering, red curve, when noise density exceeds 30%.

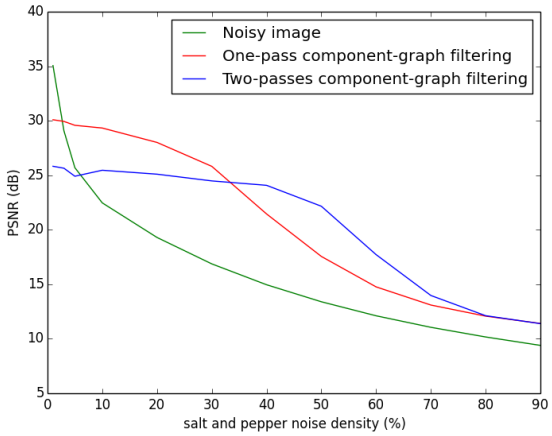


Fig. 6 Denoising PSNR on lena image for one-pass (red curve : elimination of small compact noisy structures) and two-passes (blue curve : removing of elongated noisy objects) of our proposed filtering process of shaping on component-graph, based on an elongation attribute.

Nevertheless, an intense recursive use of attribute-based filtering steps for salt-and-pepper removal can produce output artefacts, most apparent on the edges and details of the image.

6.2 Multi-modality image segmentation

Medical imaging has evolved in the past decades, with the development of new modalities, the increasing availability of hybrid imaging systems, and a greater integration of multiple modalities, providing complementary information. Medical imaging modalities can be divided into two major categories : anatomical, that mainly depict morphology; and functional, that describe information of the physiological and biochemical processes in the underlying anatomy. The combined processing of multi-modalities offers great potential for achieving more accurate and more reliable image segmentation.

Positron Emission Tomography (PET) using ^{18}F-FDG, provides metabolic activity visualization characterized by the intensity of an injected radiotracer (^{18}F-FDG). It is routinely used in cancer imaging for diagnosis and characterization of malignant tissues, corresponding to FDG hyperfixations (represented as black regions surimposed in purple on Figure 9(b)). PET images are classically associated to anatomical X-ray computed tomography (CT) images (Figure 9(a)), for visualizing the anatomy. The interpretation of PET images require a thorough knowledge of the normal patterns of the FDG uptake. Indeed, as FDG uptake reflects glucose metabolism, an increased FDG uptake is not limited to malignant tissues. In addition to this abnormal uptake, associated to tumours, both physiological and normal FDG uptakes are seen in various organs such as the brain, the heart, the liver, the bladder,... Then, such coupled images can provide complementary information : when abnormal uptake on PET helps the clinicians to identify subtle anatomical changes of potential cancer lesions in lightly contrasted areas on CT images where the tumour would not be detected, CT enhances specificity by localising FDG updates corresponding to physiological sites. It is then pertinent to process them as a unique bivalued image in order to more accurately extract the lesions and their activity. The idea is to highlight tumours, i.e. maximal intensities in the PET image (represented as black regions in the PET) and discriminate those corresponding to physiological uptakes, using the CT information.

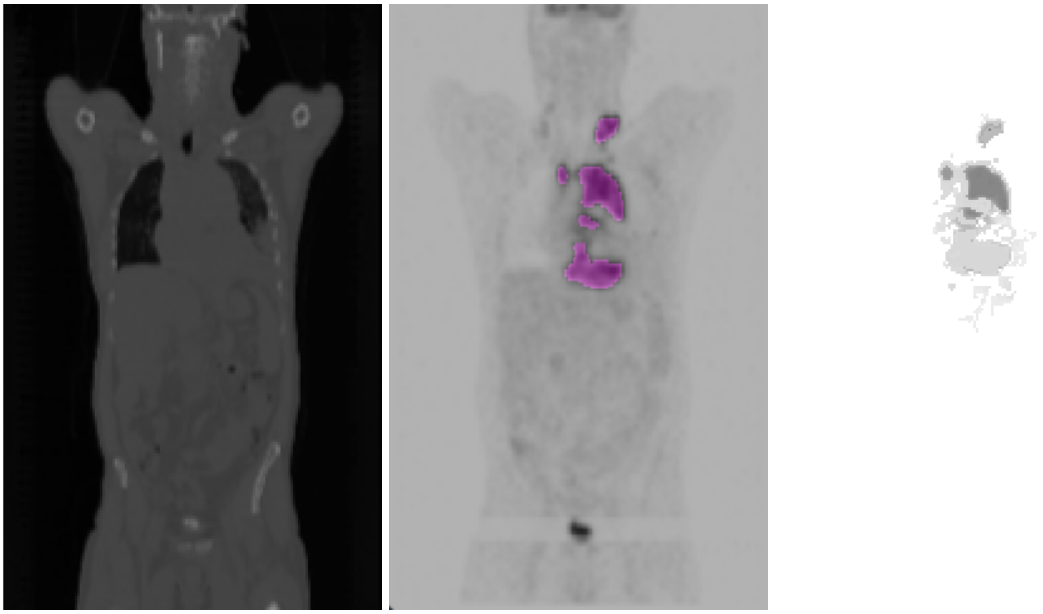
In contrast to PET images, where the canonical order on $\mathbb{R}$ captures the semantics of metabolic activity, this order is – partially – meaningless with respect to the Hounsfield scale in CT. Consequently, we apply a non injective mapping on CT to the regions known as physiological uptakes on the PET. More precisely, we associate the lowest CT values to tissues of extremal (low, e.g., water and blood; and high, e.g., bones) intensities. The order $\leq$ on $\mathbb{R}$ for the resulting image associates the least values in the CT data to tissues which are more likely to induce false positives in PET. The value space is subsampled to 256 values for both PET and CT, leading to a space of $\mathbb{V}$ of 65536 distinct values.

In the case of adults lymphomas, lymphatic lesions in the thorax are characterized as compact masses. Consequently, the criterion considered for lesions filtering is also the compactness factor [65]. As above, the filtered image is reconstructed in $\mathbb{V}$, following the policy proposed in Equation (17).

Results are shown in Figure 9(c). We observe satisfactory spatial accuracy for the detection of lesions (ground truth superimposed in purple on the PET image in Figure 9(b)). All lymphatic lesions are detected and we note a good discrimination between the lesions and false positives (i.e. the bladder physiological uptake). A medical study to validate the process is in progress and will be the topic of another article.

7 Conclusion

By coupling the two recently introduced notions of shaping and component-graph, we open the way to the development of new connected operators based on morphological hierarchies, to process multivalued images. This work constitutes a first algorithmic contribution to



(a) (b) (c)

Fig. 9 Coupled CT (a) and PET (b) images. (b) Ground-truth of the lesions, in purple. (c) Multivalued shape-based filtering from (a+b), visualized in the PET value space. ((a,b) Courtesy M. Meignan, Hôpitaux Universitaires Henri-Mondor, Lymphoma Academic Research Organization, Créteil, France.)

such an approach, in the field of multivalued mathematical morphology. Beyond promising results obtained on application examples, this pioneering work opens various algorithmic , methodological and applicative perspectives.

From an algorithmic point of view, we have only considered here the case of multiband images, i.e., images that take their values in well-structured ordered sets, namely complete lattices. This allowed us to "easily" reconstruct filtered images from pruned component-trees / graphs. Indeed, the existence of suprema / infima provides non-ambiguous valuation policies at each point. In order to go a step further, and also process more complex value spaces, it will be necessary to provide efficient (with respect to computational cost) and relevant (with respect to image processing) solutions for reconstructing filtered images when various non-comparable values coexist spatially after graph pruning. As well, we have only considered the simplest version of component-graph, that models only the values / nodes that are "visible" in the image. This assumption is fulfield in many cases, including those of the proposed application examples. However, there exist situations where the image processing issues require to handle more complex component-graphs, that explicitly model "non-visible" nodes. Such graphs are more complex to compute, but also much larger and then non-obvious to process. We will investigate algorithmic solutions to handle these richer component-graphs, and to include them in our proposed framework.

From a methodological point of view, we have considered scalar attributes as valuation for the first layer of component-graph. Such scalar attributes are generally defined on integers or real numbers, and then equipped with a canonical total order. The second layer of the proposed data-structure is therefore a component-tree. More generally, the explicit handling of vectorial attributes [66] at the first layer of the structure would lead to the construction of a component-graph at the second layer also. To handle that case, it would be mandatory to develop new strategies to perform the shaping operator for "graphs on graphs" instead of "trees on trees" in previous works, or "trees on graphs" as proposed here. The main difficulty will hinge on the handling of the space cost (and, by side effect, the computational complexity), for instance by using simplified data-structures, e.g., as investigated for the definition of multivalued trees of shapes [63].

From an applicative point of view, the proposed framework has already been utilised in medical imaging, for the segmentation of 3D images from multimodal acquisition devices (in our case, positron emission tomography coupled with X-ray computed tomography) [65, 17]. The main issues raised by these applications are directly related, on the one hand, to the design of a common spatial framework for images generally acquired

at different resolutions; and on the other hand, to the definition of orderings that actually model relevant information in each band. This can involved combining different modalities, which can be sometimes straightforward, but is often in fact quite complicated (for instance when combining morphological with functional images). In this context, coupling component-trees / graphs with other kinds of hierarchies (trees of shapes, binary partition trees, or watershed hierarchies) may constitute an interesting perspective.

In the spirit of reproducible research, the code used for the experiments of this paper is freely available at `https://github.com/bnaegel/component-graph.git`.

Acknowledgements This research was partially funded by the French *Agence Nationale de la Recherche* (Grant Agreement ANR-10-BLAN-0205) and the *Programme d'Investissements d'Avenir* (LabEx Bézout, ANR-10-LABX-58). The authors thank M. Meignan (Hôpitaux Universitaires Henri-Mondor, Lymphoma Academic Research Organisation, Créteil, France), for providing the PET / CT images.

References

1. Serra, J. (ed.): Image Analysis and Mathematical Morphology, II: Theoretical Advances. Academic Press, London (1988)
2. Najman, L., Talbot, H. (eds.): Mathematical Morphology: From Theory to Applications. ISTE/J. Wiley & Sons (2010)
3. Heijmans, H.J.A.M.: Theoretical aspects of gray-level morphology. IEEE Transactions on Pattern Analysis and Machine Intelligence **13**(6), 568–582 (1991)
4. Goutsias, J., Heijmans, H.J.A.M., Sivakumar, K.: Morphological operators for image sequences. Computer Vision and Image Understanding **62**(3), 326–346 (1995)
5. Aptoula, E., Lefèvre, S.: A comparative study on multivariate mathematical morphology. Pattern Recognition **40**(11), 2914–2929 (2007)
6. Serra, J.: Connectivity on complete lattices. Journal of Mathematical Imaging and Vision **9**(3), 231–251 (1998)
7. Talbot, H., Evans, C., Jones, R.: Complete ordering and multivariate mathematical morphology. In: ISMM, International Symposium on Mathematical Morphology, Proceedings, pp. 27–34. Kluwer (1998)
8. Angulo, J., Chanussot, J.: Color and multivariate images. Mathematical Morphology: From Theory to Applications pp. 291–321 (2013)
9. Ronse, C., Agnus, V.: Morphology on label images: Flat-type operators and connections. Journal of Mathematical Imaging and Vision **22**(2), 283–307 (2005)
10. Chevallier, E., Chevallier, A., Angulo, J.: N-ary mathematical morphology. In: ISMM, International Symposium on Mathematical Morphology, Proceedings, *Lecture Notes in Computer Science*, vol. 9082, pp. 339–350. Springer (2015)
11. Barnett, V.: The ordering of multivariate data. Journal of the Royal Statistical Society: Series A (Statistics in Society) **139**(3), 318–354 (1976)
12. Angulo, J.: Morphological colour operators in totally ordered lattices based on distances: Application to image filtering, enhancement and analysis. Computer Vision and Image Understanding **107**(1–2), 56–73 (2007)
13. Gimenez, D., Evans, A.N.: An evaluation of area morphology scale-spaces for colour images. Computer Vision and Image Understanding **110**(1), 32–42 (2008)
14. Angulo, J.: Geometric algebra colour image representations and derived total orderings for morphological operators–Part I: Colour quaternions. Journal of Visual Communication and Image Representation **21**(1), 33–48 (2010)
15. Velasco-Forero, S., Angulo, J.: Supervised ordering in R^{p}: Application to morphological processing of hyperspectral images. IEEE Transactions on Image Processing **20**(11), 3301–3308 (2011)
16. Aptoula, E., Lefèvre, S.: On lexicographical ordering in multivariate mathematical morphology. Pattern Recognition Letters **29**(2), 109–118 (2008)
17. Grossiord, É., Naegel, B., Talbot, H., Passat, N., Najman, L.: Shape-based analysis on component-graphs for multivalue image processing. In: ISMM, International Symposium on Mathematical Morphology, Proceedings, *Lecture Notes in Computer Science*, vol. 9082, pp. 446–457. Springer (2015)
18. Mazo, L., Passat, N., Couprie, M., Ronse, C.: Paths, homotopy and reduction in digital images. Acta Applicandae Mathematicae **113**(2), 167–193 (2011)
19. Rosenfeld, A.: Connectivity in digital pictures. Journal of the ACM **17**(1), 146–160 (1970)
20. Kovalevsky, V.A.: Finite topology as applied to image analysis. Computer Vision, Graphics, and Image Processing **46**(2), 141–161 (1989)
21. Kong, T.Y., Rosenfeld, A.: Digital topology: Introduction and survey. Computer Vision, Graphics, and Image Processing **48**(3), 357–393 (1989)
22. Ronse, C.: Set-theoretical algebraic approaches to connectivity in continuous or digital spaces. Journal of Mathematical Imaging and Vision **8**(1), 41–58 (1998)
23. Braga-Neto, U., Goutsias, J.: Connectivity on complete lattices: New results. Computer Vision and Image Understanding **85**(1), 22–53 (2002)
24. Braga-Neto, U., Goutsias, J.K.: A theoretical tour of connectivity in image processing and analysis. Journal of Mathematical Imaging and Vision **19**(1), 5–31 (2003)
25. Ouzounis, G.K., Wilkinson, M.H.F.: Mask-based second-generation connectivity and attribute filters. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(6), 990–1004 (2007)
26. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. IEEE Transactions on Image Processing **7**(4), 555–570 (1998)
27. Najman, L., Couprie, M.: Building the component tree in quasi-linear time. IEEE Transactions on Image Processing **15**(11), 3531–3539 (2006)
28. Berger, C., Géraud, T., Levillain, R., Widynski, N., Baillard, A., Bertin, E.: Effective component tree computation with application to pattern recognition in astronomical imaging. In: ICIP, International Conference on Image Processing, Proceedings, pp. 41–44 (2007)
29. Wilkinson, M.H.F., Gao, H., Hesselink, W.H., Jonker, J.E., Meijster, A.: Concurrent computation of attribute filters on shared memory parallel machines. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(10), 1800–1813 (2008)

30. Carlinet, E., Géraud, T.: A comparative review of component tree computation algorithms. IEEE Transactions on Image Processing **23**(9), 3885–3895 (2014)
31. Jones, R.: Connected filtering and segmentation using component trees. Computer Vision and Image Understanding **75**(3), 215–228 (1999)
32. Alajlan, N., Kamel, M.S., Freeman, G.H.: Geometry-based image retrieval in binary image databases. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(6), 1003–1013 (2008)
33. Urbach, E.R., Roerdink, J.B.T.M., Wilkinson, M.H.F.: Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(2), 272–285 (2007)
34. Westenberg, M.A., Roerdink, J.B.T.M., Wilkinson, M.H.F.: Volumetric attribute filtering and interactive visualization using the max-tree representation. IEEE Transactions on Image Processing **16**(12), 2943–2952 (2007)
35. Naegel, B., Wendling, L.: A document binarization method based on connected operators. Pattern Recognition Letters **31**(11), 1251–1259 (2010)
36. Guigues, L., Cocquerez, J.P., Le Men, H.: Scale-sets image analysis. International Journal of Computer Vision **68**(3), 289–317 (2006)
37. Passat, N., Naegel, B., Rousseau, F., Koob, M., Dietemann, J.L.: Interactive segmentation based on component-trees. Pattern Recognition **44**(10–11), 2539–2554 (2011)
38. Kiran, B.R., Serra, J.: Braids of partitions. In: ISMM, International Symposium on Mathematical Morphology, Proceedings, *Lecture Notes in Computer Science*, vol. 9082, pp. 217–228. Springer (2015)
39. Breen, E.J., Jones, R.: Attribute openings, thinnings, and granulometries. Computer Vision and Image Understanding **64**(3), 377–389 (1996)
40. Passat, N., Naegel, B.: Component-hypertrees for image segmentation. In: ISMM, International Symposium on Mathematical Morphology, Proceedings, *Lecture Notes in Computer Science*, vol. 6671, pp. 284–295. Springer (2011)
41. Perret, B., Cousty, J., Tankyevych, O., Talbot, H., Passat, N.: Directed connected operators: Asymmetric hierarchies for image filtering and segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(6), 1162–1176 (2015)
42. Naegel, B., Passat, N.: Component-trees and multi-value images: A comparative study. In: ISMM, International Symposium on Mathematical Morphology, Proceedings, *Lecture Notes in Computer Science*, vol. 5720, pp. 261–271. Springer (2009)
43. Passat, N., Naegel, B.: An extension of component-trees to partial orders. In: ICIP, International Conference on Image Processing, Proceedings, pp. 3981–3984 (2009)
44. Passat, N., Naegel, B.: Component-trees and multivalued images: Structural properties. Journal of Mathematical Imaging and Vision **49**(1), 37–50 (2014)
45. Kurtz, C., Naegel, B., Passat, N.: Connected filtering based on multivalued component-trees. IEEE Transactions on Image Processing **23**(12), 5152–5164 (2014)
46. Naegel, B., Passat, N.: Colour image filtering with component-graphs. In: ICPR, International Conference on Pattern Recognition, Proceedings, pp. 1621–1626 (2014)
47. Naegel, B., Passat, N.: Toward connected filtering based on component-graphs. In: ISMM, International Symposium on Mathematical Morphology, Proceedings, *Lecture Notes in Computer Science*, vol. 7883, pp. 350–361. Springer (2013)
48. Salembier, P., Serra, J.: Flat zones filtering, connected operators, and filters by reconstruction. IEEE Transactions on Image Processing **4**(8), 1153–1160 (1995)
49. Heijmans, H.J.A.M.: Connected morphological operators for binary images. Computer Vision and Image Understanding **73**(1), 99–120 (1999)
50. Salembier, P., Wilkinson, M.H.F.: Connected operators: A review of region-based morphological image processing techniques. IEEE Signal Processing Magazine **26**(6), 136–157 (2009)
51. Cousty, J., Najman, L., Perret, B.: Constructive links between some morphological hierarchies on edge-weighted graphs. In: ISMM, International Symposium on Mathematical Morphology, Proceedings, *Lecture Notes in Computer Science*, vol. 7883, pp. 86–97. Springer (2013)
52. Najman, L., Cousty, J.: A graph-based mathematical morphology reader. Pattern Recogntion Letters **47**, 3–17 (2014)
53. Najman, L., Schmitt, M.: Geodesic saliency of watershed contours and hierarchical segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **18**(12), 1163–1173 (1996)
54. Montanvert, A., Meer, P., Rosenfeld, A.: Hierarchical image analysis using irregular tessellations. IEEE Transactions on Pattern Analysis and Machine Intelligence **13**(4), 307–316 (1991)
55. Soille, P.: Constrained connectivity for hierarchical image partitioning and simplification. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(7), 1132–1145 (2008)
56. Salembier, P., Garrido, L.: Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. IEEE Transactions on Image Processing **9**(4), 561–576 (2000)
57. Ouzounis, G.K., Soille, P.: Pattern spectra from partition pyramids and hierarchies. In: ISMM, International Symposium on Mathematical Morphology, Proceedings, *Lecture Notes in Computer Science*, vol. 6671, pp. 108–119. Springer (2011)
58. Yau, M.M., Srihari, S.N.: A hierarchical data structure for multidimensional digital images. Communications of the ACM **26**(7), 504–515 (1983)
59. Perret, B., Lefèvre, S., Collet, C., Slezak, E.: Hyperconnections and hierarchical representations for grayscale and multiband image processing. IEEE Transactions on Image Processing **21**(1), 14–27 (2012)
60. Monasse, P., Guichard, F.: Scale-space from a level lines tree. Journal of Visual Communication and Image Representation **11**(2), 224–236 (2000)
61. Monasse, P., Guichard, F.: Fast computation of a contrast-invariant image representation. IEEE Transactions on Image Processing **9**(5), 860–872 (2000)
62. Carlinet, E., Géraud, T.: MToS: A tree of shapes for multivariate images. IEEE Transactions on Image Processing **24**(12), 5330–5342 (2015)
63. Carlinet, E., Géraud, T.: A morphological tree of shapes for color images. In: ICPR, International Conference on Pattern Recognition, Proceedings, pp. 1132–1137 (2014)
64. Xu, Y., Géraud, T., Najman, L.: Connected filtering on tree-based shape-spaces. IEEE Transactions on Pattern Analysis and Machine Intelligence (In Press)
65. Grossiord, É., Talbot, H., Passat, N., Meignan, M., Tervé, P., Najman, L.: Hierarchies and shape-space for PET image segmentation. In: ISBI, International Symposium on Biomedical Imaging, Proceedings, pp. 1118–1121 (2015)

66. Urbach, E.R., Boersma, N.J., Wilkinson, M.H.F.: Vector attribute filters. In: ISMM, International Symposium on Mathematical Morphology, Proceedings, *Computational Imaging and Vision*, vol. 30, pp. 95–104. Springer (2005)