Research Article Open Access

Éloïse Grossiord, Benoît Naegel, Hugues Talbot, Laurent Najman, and Nicolas Passat*

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

https://doi.org/10.1515/mathm-2019-0003
Received March 29, 2018; accepted September 17, 2019

Abstract: Connected operators based on hierarchical image models have been increasingly considered for the design of efficient image segmentation and filtering tools in various application fields. 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 any grey-level or multivalued images. The notion of shaping was also introduced as a way to improve the anti-extensive filtering 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 extend 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. Although the contributions of this article are theoretical and methodological, we also provide two illustration examples that qualitatively emphasize the potential use and usefulness of the proposed paradigms for image analysis purposes.

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

MSC: 06 – Order, lattices, ordered algebraic structures; 68 – Computer science

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 (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. For instance, 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], i.e. 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 elements 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

Éloïse Grossiord: Institut Universitaire du Cancer (IUCT) – Oncopôle, Toulouse, France, E-mail: eloise.grossiord@gmail.com
Benoît Naegel: ICube, Université de Strasbourg, CNRS, France, E-mail: b.naegel@unistra.fr
Hugues Talbot: CentraleSupelec, INRIA GALEN-POST, France, E-mail: hugues.talbot@centralesupelec.fr
Laurent Najman: LIGM, ESIEE, Université Paris-Est, CNRS, France, E-mail: laurent.najman@esiee.fr
***Corresponding Author: Nicolas Passat:** CReSTIC, Université de Reims Champagne-Ardenne, France, E-mail: nicolas.passat@univ-reims.fr

induced from the canonical total order on the reals or integers. In the case for multivalued images, values are not canonically equipped with a total 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). This is particularly considered for handling colour images [12–14] 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 [14, 16]), 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 induce some possible loss and bias of the 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]— we 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, 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 [19–21]; or by using morphological definitions of connectivity [6, 22–25].

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 (the most popular morphological hierarchy data structure) and previous works about the hierarchical handling of multivalued images, are proposed in Sec. 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 Sec. 3.

Our main contribution is proposed in Secs. 4 and 5. 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 describe in Sec. 4 the extension of the classical anti-extensive filtering paradigm based on component-trees; and we provide a complete algorithmic description of the way to actually implement this filtering in the case of images taking their values in multiband spaces. Two application examples, provided in Sec. 6, qualitatively illustrate the methodological interest of our hierarchical approach, for handling multivalued images and for fusing multiple binary segmentations, via component-graphs and shapings.

2 Related works

2.1 Component-trees

The component-tree is a compact, lossless, hierarchical representation of grey-level images. Induced by the inclusion relation between the binary components of successive level-sets, 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-tree in quasi-linear time, in sequential [26–28] and in distributed ways [29]. A recent survey on the different computation algorithms can be found in [30]. More generally, a recent survey on partition hierarchies, including the component-trees, can be found in [31].

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 [25, 26, 32] 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 [33], classification [34], interactive visualisation [35], or document binarisation [36].

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 compute an optimal cut [37], interpreted as a segmentation of the image. This often formulated as an optimization problem, where the space of solutions is composed of partitions from the hierarchy (in this context, a notion of braids of partitions [39] was introduced as a general framework for the optimization of segmentation based on hierarchical partitions). This is the basis for interactive segmentation [38]. The second approach consists of determining locally the nodes that should be preserved or discarded, based on attribute values [40] 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, 32], recalled in Sec. 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 or grey-level 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 [41] to deal with ordered families of connectivities, leading to component-hypertrees, and in [42] 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 [43]. Then, actual extensions of component-trees to partially-ordered value images were pioneered in [44] and further formalized in [45]. Except in specific cases where the values are themselves hierarchically organized [46], 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 [47, 48].

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, 26, 49, 50]. In this context, operators based on hierarchical image models, i.e. trees, have been the object of several structural, algorithmic and methodological developments [51] in order to tackle issues associated to specific application fields.

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

The first rely on hierarchical clustering. Indeed, hierarchical segmentation trees aim at either growing and merging regions in a bottom-up fashion, 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. In other words, they 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.

Alternatively, morphological trees focus on the inclusion relationship between components. These components result from thresholding operations on the image at every levels. A total order on the image values ensures the inclusion of the level sets. Such total ordering is then required to compute these trees. Unfortunately, and contrary to grey-level images, the spaces in which multivalued images take their values are not canonically equipped with total orders. Then, 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 self-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 restrictive constraints on the value space [46], or a final simplification. More specifically, for the extension of the tree of shapes to multivalued image, it was proposed to marginally compute the tree of shapes for each colour channel, and merge them into a single tree [62]. But then, the merging decision does not rely on values anymore but on properties computed in a shape-space [63].

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 leads to a data structure that is no longer a tree, but a DAG. This is the case for the notion of component-graph [45], 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, 32]. This is in particular the case for handling spatial complexity [47], pruning policies and image reconstruction [48].

Recently, a new notion of shaping [64] was introduced as an efficient way to improve the framework of anti-extensive filtering of [26, 32], by considering a two-layer component-tree for grey-level image processing [64, 65]. In [17], we proposed to associate both 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 Background notions

This article is set in the framework of vertex-valued graphs. We 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 Secs. 4 and 5, how to carry out shaping on component-graphs to handle multivalued images. For the sake of clarity, Sec. 3 is written in a self-contained 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 $\exists x, y \in \Gamma, (x \nleq y \wedge y \nleq x)$). The word “partial” indicates that there is no guarantee that two elements can be compared

An ordered set can be modelled via its Hasse diagram, which depicts the covering 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, illustrations are drawn so that elements are placed 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 (i.e. a set $\mathcal{P}$ of nonempty disjoint subsets of Γ whose union is Γ).

Let $\mathcal{F} : \Gamma \to \mathbb{V}$ be a function such that $\mathbb{V}$ is a nonempty 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.

3.3 Component-tree

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 $\perp$.

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

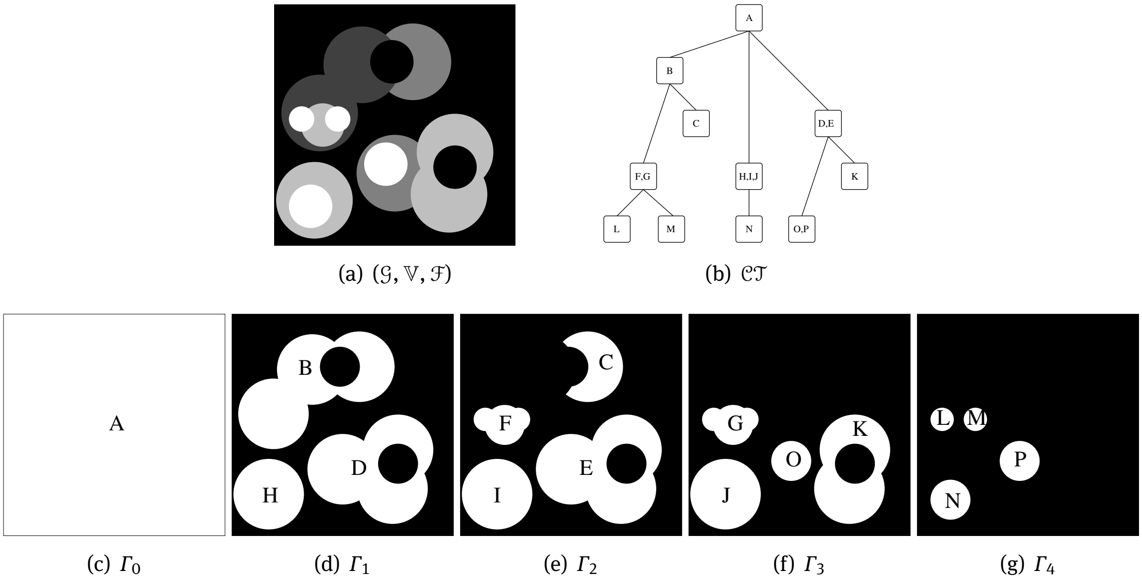
$$\Gamma_v = \{x \in \Gamma \mid v \leq \mathcal{F}(x)\} \tag{1}$$

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 [26] 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 $v \in \mathbb{V}$. Then, 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 $\mathcal{F}$.



(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

Figure 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 ≤. (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 in (c–g).

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 $\mathcal{G} = \mathcal{G}_\perp$; 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 Fig. 1). The order relation ≤ between nodes defines a parenthood relationship: a node $X_1 \in \Psi$ is the parent of a node $X_2 \in \Psi$ if $X_2 \subset X_1$ and if there is no other node $X_3 \in \Psi$ such that $X_2 \subset X_3 \subset X_1$. In that case, we also say that X_2 is a child of X_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

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 $\perp$. The graph $\mathcal{G}$ also remains connected, but from now on the order relation ≤ 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 component of the threshold set Γ_v inducing $\mathcal{G}_v$ at value v. (Contrary 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 the order relation $\trianglelefteq$ on the valued connected components of Θ as

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

This order[1], 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 [45] $\mathcal{CG}$ associated to 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, either these parent nodes meet without inclusion, i.e. they are not mutually included, or one of them is included in the other.

The component-graph still has a unique greatest node that is the maximum for the Hasse diagram, namely $(\Gamma, \perp)$; 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, relying on different subsets of valued connected components:

1. Θ represents all the valued connected components induced by $\mathcal{G}$;
2. $\dot{\Theta}$ 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. $\ddot{\Theta}$ gathers the valued connected components which are generators of $\mathcal{F}$, i.e. those that actually contribute to the definition of the image support.

$$\dot{\Theta} = \{(X, v) \in \Theta \mid \forall (X, v') \in \Theta, v \nless v'\} \tag{5}$$

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

Based on these definitions, we observe that

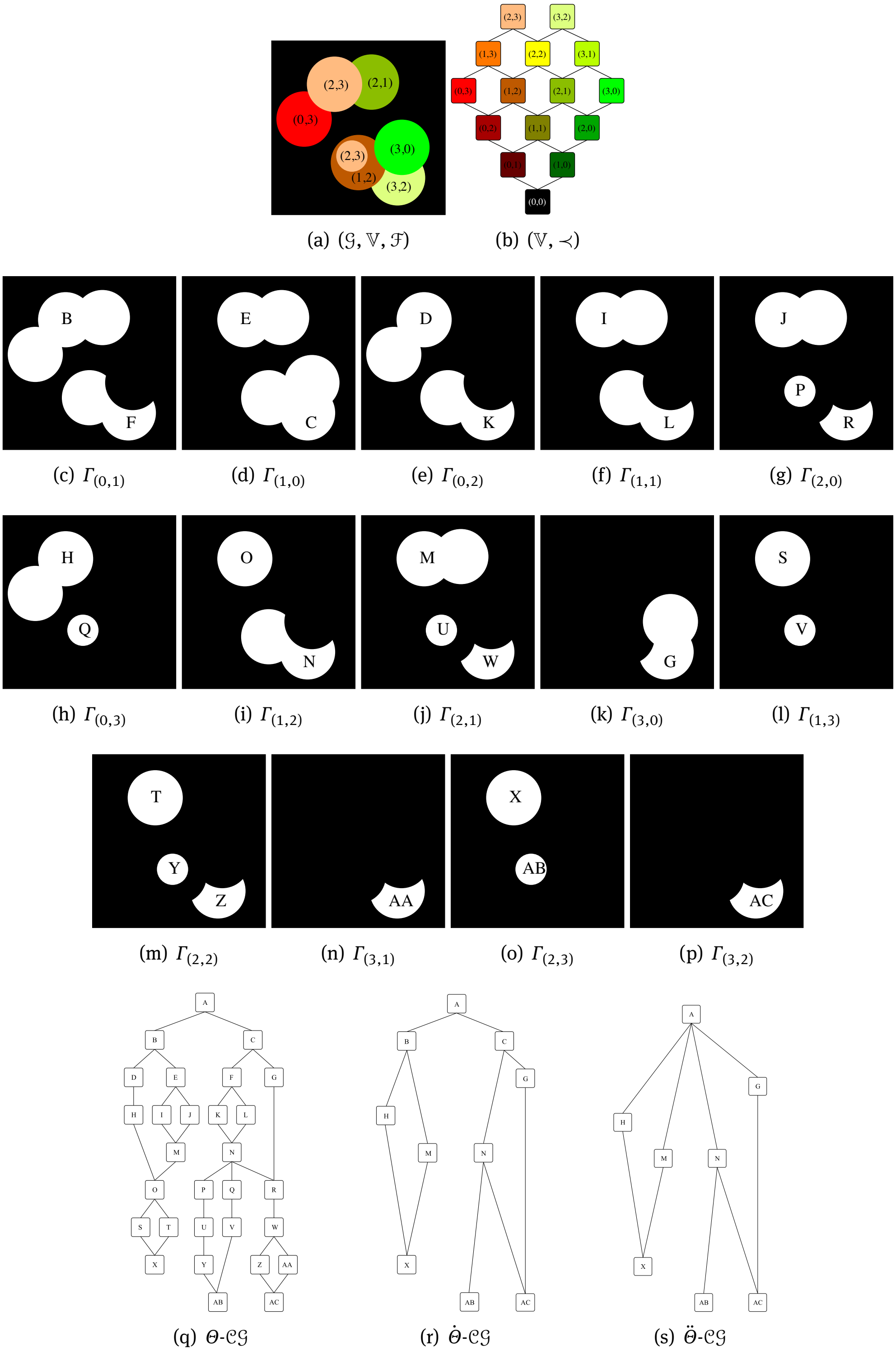
$$\ddot{\Theta} \subseteq \dot{\Theta} \subseteq \Theta \tag{7}$$

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 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 directly depends on the targeted image processing application. For instance, when $(\mathbb{V}, \leq)$ is a hierarchy of concepts, e.g. an ontology, it is mandatory to compute all the nodes for a complete description of the semantics of the image [46]; then the Θ-component-graph is relevant. By contrast, when performing anti-extensive filtering on colour images, it is important to avoid the appearance, in the result, of colours non-present in the input data [47]; then the $\ddot{\Theta}$-component-graph is relevant.

An example of multivalued image $\mathcal{F}$ and its associated value set $\mathbb{V}$ are provided in Fig. 2(a) and (b). Fig. 2(c–p) depicts the various valued connected components obtained from this image. More precisely, the

[1] Practically, when $\leq$ is a total order, the component-graph and the component-tree are isomorphic [45]. 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 Eq. (10).



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

(c) $\Gamma_{(0,1)}$ (d) $\Gamma_{(1,0)}$ (e) $\Gamma_{(0,2)}$ (f) $\Gamma_{(1,1)}$ (g) $\Gamma_{(2,0)}$

(h) $\Gamma_{(0,3)}$ (i) $\Gamma_{(1,2)}$ (j) $\Gamma_{(2,1)}$ (k) $\Gamma_{(3,0)}$ (l) $\Gamma_{(1,3)}$

(m) $\Gamma_{(2,2)}$ (n) $\Gamma_{(3,1)}$ (o) $\Gamma_{(2,3)}$ (p) $\Gamma_{(3,2)}$

(q) Θ-$\mathfrak{CG}$ (r) $\dot{\Theta}$-$\mathfrak{CG}$ (s) $\ddot{\Theta}$-$\mathfrak{CG}$

Figure 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 $(\mathbb{V}, \leq)$. 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 X of value $(2,3)$), or "partially visible" (e.g. the green G of value $(3,1)$), or "totally hidden" (e.g. the brown B of value $(0,1)$). Those that are either partially or totally visible participate to the formation of the image support and then belong to Θ, $\dot{\Theta}$, $\ddot{\Theta}$. 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.

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 Fig. 2(q–s).

The component-graph is a relevant extension of the component-tree, as (i) both data structures are compliant for totally 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 Eq. (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 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: Theory

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 $\mathbb{V}$ possibly 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:

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

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 images with several bands of values; we then have $\mathbb{V} = \mathbb{V}_1 \times \ldots \times \mathbb{V}_n$, and this Cartesian product induces 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 $X \subseteq \Omega$ and any $v \in \mathbb{V}$, we define the cylinder function $C_{(X,v)}$ of support X and value v, as:

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

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 Eq. (10). More precisely, the image can be expressed as the point-wise supremum of the nodes of its associated component-tree (Ψ) or component-graph (Θ):

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

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 $\widehat{\Psi} \subseteq \Psi$. Then, it is possible to filter the image $\mathcal{I}$ by discarding some of the nodes of its hierarchical representation, and reconstructing a resulting image from the preserved nodes. Each point $x \in \Omega$ in the filtered image 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, 32]. 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 $\widehat{\mathcal{CT}}$.

$$\begin{tikzcd}
(\mathcal{S}, \mathbb{V}, \mathcal{I}) \arrow[r, "(i)"] \arrow[d] & (\mathcal{CT}, \mathbb{R}, \mathcal{V}) \arrow[d, "(ii)"] \\
(\mathcal{S}, \mathbb{V}, \widehat{\mathcal{I}}) & (\widehat{\mathcal{CT}}, \mathbb{R}, \mathcal{V}_{|\widehat{\Psi}}) \arrow[l, "(iii)"']
\end{tikzcd} \tag{11}$$

Step (i) is carried out from a wide range of available component-tree construction methods [30], while Step (iii) is straightforward from Eq. (10).

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 (i) a selection criterion, i.e. a Boolean predicate related to the valuation $\mathcal{V} : \Psi \to \mathbb{R}$ that indicates if a node satisfies a required property; and (ii) a reduction policy to determine which parts of the component-tree should be kept or removed. The nature of the valuation $\mathcal{V}$ guides the decision of preserving or discarding a node. If $\mathcal{V}$ models an increasing attribute, the removal of a node implies that of all its descendants. Contrary, if $\mathcal{V}$ models a non-increasing attribute, some rejected nodes can have preserved descendants. In other words, 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 for handling such situation, including in particular the Min, Direct and Max ones [26, 32]:

- 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 and 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 preserves exactly the nodes fulfilling the criterion, but it 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 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 Eq. (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 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, 32]:

$$\begin{tikzcd}
(\mathcal{S}, \mathbb{V}, \mathcal{I}) \arrow[r, "(i)"] \arrow[d] & (\mathcal{CG}, \mathbb{R}, \mathcal{A}) \arrow[d, "(ii)"] \\
(\mathcal{S}, \mathbb{V}, \widehat{\mathcal{I}}) & (\widehat{\mathcal{CG}}, \mathbb{R}, \mathcal{A}_{|\widehat{\Theta}}) \arrow[l, "(iii)"']
\end{tikzcd} \tag{12}$$

However, due to the more complex structure of multivalued images and component-graphs, extending this framework is not straightforward. In particular, it raises two difficulties. First, 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. Second, Step (*iii*) becomes an ill-posed problem, depending on the nature of the order $\leq$ and the organization of 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 selection, the reconstruction becomes subject to arbitrary decisions, due to non-determinism at the parent nodes intersection. In addition, structural stability may be lost as the result image can 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

The paradigm of shaping [64] proposes to extend 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.

The first (inner) layer corresponds to the component-tree $\mathcal{CT}$ of the image. The paradigm of shaping is to consider this component-tree $\mathcal{CT}$ itself as an image, and to build a second (outer) layer of component-tree from it. Indeed, from a functional point of view, a component-tree 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 is the parent of the other. This approach is tractable only if the space of 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 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” $\mathcal{CT}'$ is 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, 32], 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 attributes computed from the nodes of $\mathcal{CT}$ are 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.

$$\begin{tikzcd}
(\mathcal{S}, \mathbb{V}, \mathcal{I}) \arrow[r, "(i)"] \arrow[d] & (\mathcal{CT}, \mathbb{R}, \mathcal{V}) \arrow[r, "(i')"] & (\mathcal{CT}', \mathbb{R}, \mathcal{V}) \arrow[d, "(ii)"] \\
(\mathcal{S}, \mathbb{V}, \widehat{\mathcal{I}}) & (\widehat{\mathcal{CT}}, \mathbb{R}, \mathcal{V}_{|\widehat{\Psi}}) \arrow[l, "(iii)"'] & (\widehat{\mathcal{CT}}', \mathbb{R}, \mathcal{V}_{|\widehat{\Psi}'}) \arrow[l, "(iii')"']
\end{tikzcd} \tag{13}$$

The main limitation of this framework is that it considers a tree as intermediate data structure, thus limiting its use to grey-level images.

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; 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 also 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.

$$\begin{tikzcd}
(\mathcal{S}, \mathbb{V}, \mathcal{I}) \arrow[r, "(i)"] \arrow[d] & (\mathcal{CG}, \mathbb{R}, \mathcal{A}) \arrow[r, "(i')"] & (\mathcal{CT}, \mathbb{R}, \mathcal{A}) \arrow[d, "(ii)"] \\
(\mathcal{S}, \mathbb{V}, \widehat{\mathcal{I}}) & (\widehat{\mathcal{CG}}, \mathbb{R}, \mathcal{A}_{|\widehat{\Theta}}) \arrow[l, "(iii)"'] & (\widehat{\mathcal{CT}}, \mathbb{R}, \mathcal{A}_{|\widehat{\Psi}}) \arrow[l, "(iii')"']
\end{tikzcd} \tag{14}$$

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 have to be tackled, in particular for the two reconstruction steps (iii), from the component-tree to the component-graph and then to the image. In Sec. 5, we propose some algorithmic solutions to these issues.

5 Shape-space analysis of multivalued images: Algorithmics

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. 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 in priority.

5.1 Component-graph construction

Three variants of component-graphs were introduced in [45] (see Sec. 3.4), in particular to simplify $\mathcal{CG}$ by considering smaller subsets of Θ. In Step (i) of our framework, that builds the first layer component-graph $\mathcal{CG}$ from the multivalued image $(\mathcal{S}, \mathbb{V}, \mathcal{I})$, we will often choose to consider the lightest version $(\ddot{\Theta})$ of component-graph (Fig. 2(s)), i.e. the one that represents only the nodes which actually contribute to the construction of the image according to Eq. (10). In the case of $\ddot{\Theta}$, we use the component-graph construction algorithm proposed in [47]. (However, an algorithm was recently proposed in [66, 67] for building, more generally, the component-graphs Θ; then, the proposed framework is, of course, also tractable for such "complete" component-graphs.) For the sake of readability, we will now note $\ddot{\Theta}$ as Θ.

The choice of working with the $\ddot{\Theta}$-component-graphs 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 with semantic-content images considered for instance in [46]). The $\ddot{\Theta}$-component-graph is the only which guarantees that no new value will be introduced via a node initially hidden in the graph; this is a reasonable property in this context. Experimentally, it was observed in [47] that such component-graph was indeed relevant for filtering (denoising, simplification) purposes.

Second, from a space complexity point of view, this component-graph is 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.

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

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 generally 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 generally have $\mathcal{A} : 2^{\Omega} \to \mathbb{R}$;
(3) *structural* properties, relying on the relationship between the node and its neighbourhood within the component-graph. By contrast with spectral and geometric properties, that describe the node locally (i.e. on its own support), structural properties describe the node with respect to its environment. For instance, relevant information can be related to branch length, position within the graph, number of children / parent nodes, etc. The definition of such properties then requires to observe the component-graph in a local or semi-global way (generally around the node of interest);

or any combination of some of these three classes. In the case of multiple criteria, standard linear combinations can be considered (mean, weighed functions), but also nonlinear combinations (min, max, median). The main difficulty then consists of making the criteria values relevantly comparable; this generally requires a normalization step.

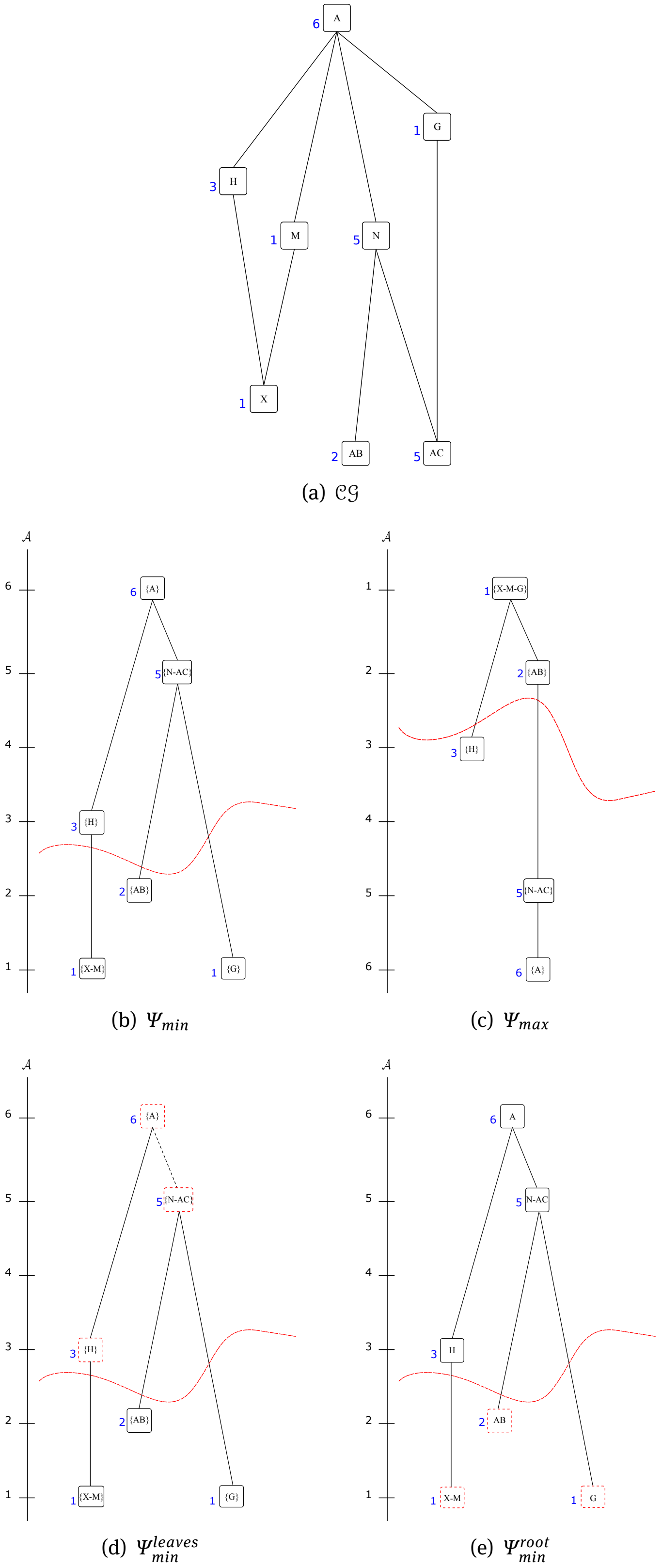
Contrary to the other kinds of component-graphs, the chosen version of $\mathcal{CG}$ is relatively light. As a consequence, a criterion of type (3) would be weakly relevant, as all “internal” nodes of the graph are not modelled, thus making the graph structurally sparse. Indeed, the elimination of nodes 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 level semantics— 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. Indeed, the component-graph already models the order between values in $\mathbb{V}$ via its structure. The spectral handling then occurs 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 $\mathcal{V}$ from which will be processed the tree pruning. In order to preserve the good properties of such filtering, it is essential that this new attribute $\mathcal{V}$ keep 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 values A_l of the leaves of its branch. Such criterion remains increasing, thus authorizing 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. This tree is the data structure that will be considered for the pruning process (Step (*ii*)).



(a) $\mathcal{CG}$

(b) Ψ_{min}

(c) Ψ_{max}

(d) Ψ_{min}^{leaves}

(e) Ψ_{min}^{root}

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

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 (for segmentation) or structures to be removed (for filtering or denoising).

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 (Fig. 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 (Fig. 3(c)) is adopted. The inversion of the attribute allows one to switch from a representation to the other.

Once the min-/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), i.e. when structures of interest have to be preserved, relevant nodes are selected by preserving the distal parts, i.e. the branches of the tree (Fig. 3(d)). In the second case (filtering), i.e. when structures of interest have to be eliminated, the pruning consists of removing those distal parts of the tree, and preserving the proximal part (Fig. 3(e)). In each case, 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 of this node. Following the above classification of properties (spectral, geometric, structural, see Sec. 5.2), this new valuation $\mathcal{A}$ —that was obtained from a valuation based on geometric properties— is now a valuation based on spectral properties in the context of shaping, since the attribute of the component-graph becomes the valuation of the component-tree, 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 $\{K_i = (X_i, v_i)\}_{i=1}^{k} \subseteq \Theta$, with $k \geq 1$.

Once the pruning of the component-tree has reduced the number of nodes Y preserved in $\widehat{\Psi}$, two main ways 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 $\widehat{\Theta}$; see Fig. 4.

A first way is to keep all the K_i contributing to the preserved Y, since they all provide spectral information in the image.

But since spectral information is already taken into consideration within the inner layer component-graph, a second way is to exclusively consider spatial information carried by the preserved nodes Y. Indeed, 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 $\trianglelefteq$ (and the $\subseteq$) relation. When dealing with geometric criteria, only these latter —maximal— nodes, that contribute to define the support $\bigcup_{i=1}^{k} X_i$ of Y in Ω, are of interest. In other words, if Y is preserved in $\widehat{\Psi}$, only the nodes $K_i = (X_i, v_i)$ of Y such that X_i is a maximal subset of Ω within Y 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.

Then, 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[2] of the nodes $K \in \Theta$ belong to $\widehat{Y}$ for at least one $Y \in \Psi$. In other

[2] Erratum: 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 always true. 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

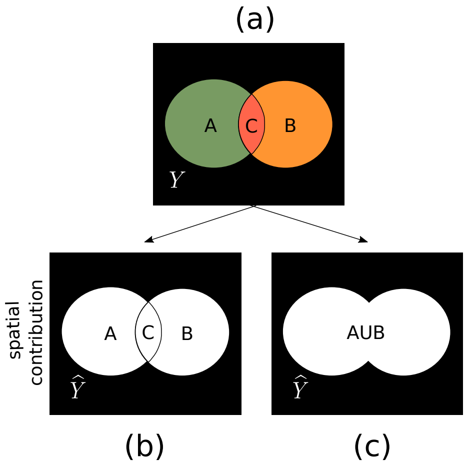


Figure 4: Component-graph filtering. (a) A node Y from the outer layer component-tree preserved in $\widehat{\Psi}$ after tree pruning. This node Y is composed of three nodes of the component-graph $\mathfrak{CG}$: A, B and C such that C ⊂ A and C ⊂ B. Two choices can be made for the filtering of Y: either (b) 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 $A \cup B$ defines the spatial support of Y while C does not participate to the definition of the boundaries of Y.

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 minimizing the risks of 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 the support of any $Y \in \Psi$ from a single node $K \in \Theta$, while the second can require several nodes of Θ since values of $\mathbb{V}$ may be non-comparable.

5.5 Image filtering

Our data structure considered for processing an image is composed of two layers of component-graph / tree. Then, the final filtering of the image has to go successively through these two layers, 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 $Y \in \widehat{\Psi}$.
(2) A given node $K \in \Theta$ may belong to $\widehat{Y}_j$, for several nodes $Y_j \in \widehat{\Psi}$. Then, the second step —which leads to the final reconstruction of the image— handles the conflicting intersections between those regions Y_j.

We recall that we deal with multivalued images, and we assume that the value space $(\mathbb{V}, \leq)$ is structured as a complete lattice. As a consequence, any subset of values 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, 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 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:

degenerated component-tree $\mathfrak{CT}$ composed by a single node. However, for "reasonable" 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 $Y \in \Psi$.

- v can be set as the *supremum* of all the v_i of each node $K_i = (X_i, v_i) \in \widehat{Y}$. This policy 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 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 value associated to all these nodes can be relevantly considered. However, 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 between the values assigned to nodes K belonging to several nodes $Y_j \in \widehat{\Psi}$.

(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 of K. This may create connected components valued with intensities higher than in the original image, due to the composition of two successive supremum operators. The reconstruction of the filtered image can then be formalized as:

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

- alternatively, the value finally assigned to the node K can be defined as the *infimum* of all the values in conflict. This policy tends to attenuate the effects of the initial supremum operator. In particular, the result will be closer to the initial image than with the above policy. However, it does not ensure to retrieve anti-extensivity. The reconstruction of the filtered image can be formalized as:

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

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

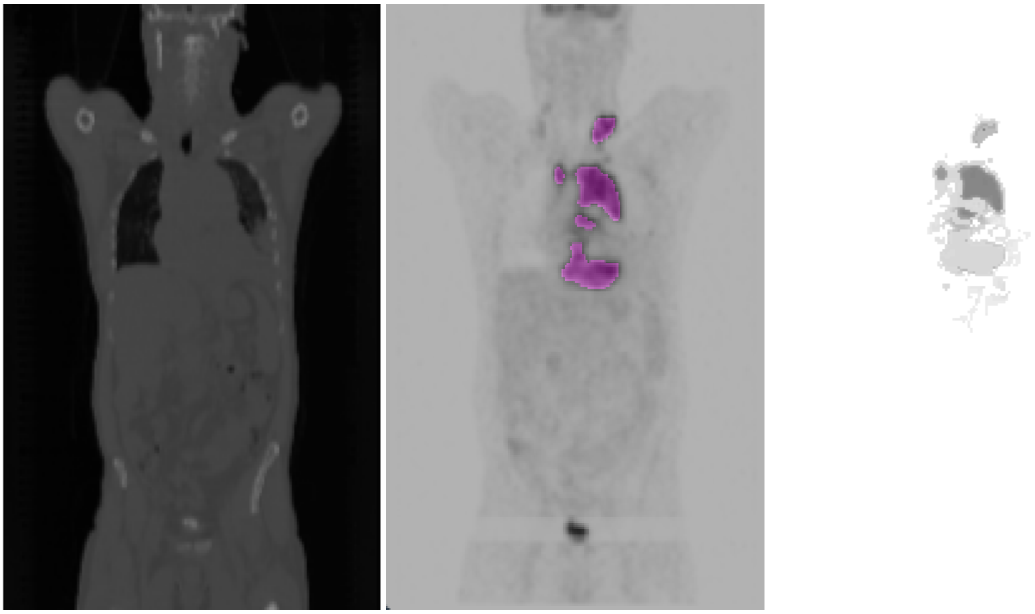
- the value finally assigned to the node K can be defined as the *supremum* of all the values in conflict. This policy is justified by the fact that a node $Y \in \Psi$, defined as the union of several nodes of Θ, should not lose its geometry in the filtered image. It allows us to offset the flattening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

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

- alternatively, the value assigned to the node K can be defined as the *infimum* of the values in conflict. This policy will tend to completely flatten the intensity in the image; 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 be formalized as:

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

In practice, there is no universally good strategy, within the four proposed above; the relevance of an approach actually depends on the purpose. However, it is important to notice that the first two, based on an



(a) (b) (c)

Figure 5: Coupled CT (a) and PET (b) images. (b) Ground-truth of the lesions, in purple. (c) Multivalued shape-based processing from (a+b), visualized in the PET value space. (a,b) Courtesy M. Meignan, Hôpitaux Universitaires Henri-Mondor, Lymphoma Academic Research Organization, Créteil, France.

initial supremum operator do guarantee neither extensivity nor anti-extensivity. Indeed, we have:

$$\widehat{\mathcal{I}}_{\vee\vee} \geq \widehat{\mathcal{I}}_{\wedge\vee} \tag{19}$$

but we can neither ensure $\widehat{\mathcal{I}}_{\vee\vee}, \widehat{\mathcal{I}}_{\wedge\vee} \geq \mathcal{I}$ nor $\leq \mathcal{I}$. By contrast, the last two strategies do guarantee anti-extensivity:

$$\mathcal{I} \geq \widehat{\mathcal{I}}_{\vee\wedge} \geq \widehat{\mathcal{I}}_{\wedge\wedge} \tag{20}$$

In particular, $\widehat{\mathcal{I}}_{\vee\wedge}$ is the closer result image that remains below $\mathcal{I}$.

These various strategies, although presented in the general case of handling all bands of a multivalued image, can be restricted to a 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 Illustrations

We now present two application cases involving shape-space analysis on component-graphs. Both deal with image segmentation, but each in a different way. The first application case deals with segmentation of multimodal images, and the component-graph is then used for modeling these modalities in a unified way. The second application case deals with segmentation fusion, and the component-graph is then used for modeling the various segmentation results as a richer alternative to fuzzy paradigms.

In both experiments, the $\ddot{\Theta}$-component-graph is used. On the one hand, it was shown in [45] that the space cost of this component-graph is $\mathcal{O}(|\Omega|)$, i.e. linear (and possibly sublinear) with respect to the size of the image support. On the other hand, it was observed in [47] that the construction of this component-graph has a computational cost $\mathcal{O}(|\Omega|^2)$, experimentally corrected to $\mathcal{O}(|\Omega|^{1,5})$.

6.1 Multimodal medical image segmentation

We consider a first application example in the context of multimodal medical imaging. (This application example was initially provided in [17]; it is presented here with more details.) The image is a function taking its values in $\mathbb{Z} \times \mathbb{N}$ with two bands corresponding each to a given imaging modality, namely morphological X-ray Computed Tomography (CT) and functional Positron Emission Tomography (PET). The segmentation of such images is designed to emphasise tumours based on their shape and metabolic activity.

Our purpose is not to prove that our results reach the state of the art for PET-CT segmentation (see [68, 69] for recent surveys on this active research topic). We aim to qualitatively emphasise that the proposed framework is versatile enough for encompassing a wide range of applications. In particular, we aim to show that it opens the way to alternative possibilities of image processing of multivalued images via connected operators.

Positron Emission Tomography visualizes metabolic activity characterized by the intensity of an injected radiotracer (here, ^{18}F-FDG). It is routinely used in cancer imaging for diagnosis and characterization of malignant tissues, corresponding to FDG hyperfixations (black regions surimposed in purple in Fig. 5(b)). PET images are classically associated to X-ray computed tomography images (Fig. 5(a)), for visualizing the anatomy.

The interpretation of PET images requires a thorough knowledge of the normal patterns of the FDG uptake. Indeed, FDG uptake reflects glucose metabolism. In particular, FDG uptakes are seen in cancerous lesions, but also in various organs such as the brain, the heart, the liver, the bladder. Then, CT images can provide complementary information by localizing FDG uptakes corresponding to physiological sites. Consequently, it is relevant 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) and discriminate those corresponding to physiological uptakes, using the CT information.

In contrast to PET images, where the canonical order $\leq$ on $\mathbb{N} = \mathbb{V}_{PET}$ captures the semantics of metabolic activity, this order is partly meaningless on $\mathbb{Z}$ with respect to the Hounsfield scale in CT. Consequently, we apply a non-injective mapping $m_H : \mathbb{Z} \to \mathbb{N}$ on CT to the regions known as physiological uptakes on the PET. More precisely, we associate the lowest values of $\mathbb{V}_{CT} = \mathbb{N}$ to tissues of extremal Hounsfield intensities. Such mapping can be defined by $m_H(v) = \max\{M - |v - K|, 0\}$ with $M > 0$ and $K \in [100, 300]$, thus providing the highest signal M for soft tissues (with Hounsfield values around $[100, 300]$), and lowest values for non-relevant tissues (for instance fat and water, around $[-100, 0]$ or bones around $[700, 1\,000]$). The order $\leq$ on $\mathbb{V}_{CT}$ 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 $\mathbb{V} = \mathbb{V}_{CT} \times \mathbb{V}_{PET}$ of $256^2 = 65\,536$ distinct values.

In the case of adults lymphomas, lymphatic lesions in the thorax are characterized as compact masses. Consequently, the criterion $\mathcal{A} : \Theta \to \mathbb{R}$ chosen for lesions filtering is the compactness factor [65]. The outer layer component-tree is then built as a max-tree with respect to the value set $\mathbb{R}$ of $\mathcal{A}$. Our purpose is here to select the nodes of highest values, i.e. in the distal parts of the branches, in a segmentation paradigm. The processed image is finally reconstructed following the policy proposed in Eq. (17).

For visualization purposes, the results are only depicted in the $\mathbb{V}_{PET}$ value band ($\mathbb{V}_{CT}$ does not present any interest in this context), see Fig. 5(c). We observe a satisfactory spatial accuracy between the detection of lesions (Fig. 5(c)) and the ground-truth (purple areas in Fig. 5(b)). In particular, all lymphatic lesions are detected, whereas potential false positives were discarded (e.g. the bladder physiological uptake).

6.2 Extended fuzzy framework for segmentation fusion

We now consider a second application case, still in the context of segmentation. However, it differs from the previous as the notion of component-graph is no longer used for modeling the structure of the value space, but for handling the fusion of many segmentation results from a spatial point of view.

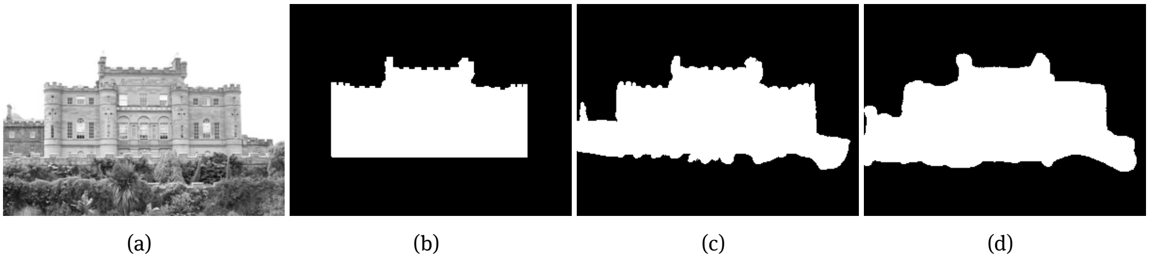


Figure 6: (a) Original image. (b–d) Manual segmentations, from the segmentation evaluation database [72].

Let us suppose that, for a given image $\mathcal{I}$ defined on a support Ω, we have $k \geq 2$ segmentation results for a same structure of interest. These k segmentation results are binary sets $S_i \subseteq \Omega$ such that for any $i \in [\![1, k]\!]$, $x \in S_i$ means that the point x belongs to the i-th segmentation result.

Our purpose is to fuse these k segmentation results. We define the function

$$\left| \begin{array}{cccc} S & : & \Omega & \to & \{0,1\}^k \\ & & x & \mapsto & (x_i)_{i=1}^k \end{array} \right. \tag{21}$$

where $x_i = 1$ (resp. 0) if $x \in S_i$ (resp. $x \notin S_i$). In other words, S provides, for each point of the image support, a k-uple of binary values that express the result of the k segmentation results at this point.

A classical way to carry out segmentation fusion is to define a fuzzy function $F : \Omega \to [0,1]$ such that for any $x \in \Omega$, we have $F(x) = \frac{1}{k}\|S(x)\|_1$, where $\|.\|_1$ is the ℓ_1 norm. In other words, we compute the probability that x belongs to the fused segmentation result, by assuming that all the k segmentation results are equiprobable. This function F is a grey-level image, and it is possible to build a standard component-tree for allowing the user to relevantly carry out local thresholding on this fuzzy fusion map. However, the function F “flattens” the vectorial information carried by the k-uples of binary values.

For improving this fuzzy framework for segmentation fusion, we consider the set $\{0,1\}^k$ as the value set $\mathbb{V}$, with a partial order $\leq$ defined as

$$(x_i)_{i=1}^k \leq (y_i)_{i=1}^k \Leftrightarrow \forall i \in [\![1,k]\!], x_i \leq y_i \tag{22}$$

Then, $(V, \leq)$ is isomorphic to the power-set of a set of k elements endowed with the inclusion relation. In particular, $(V, \leq)$ is a complete lattice.

We can then build the component-graph $\mathcal{G}$ associated to the “image” S. Each node $K = (X, (v_i)_{i=1}^k) \in \Theta$ of this component-graph, with $\|S(x)\|_1 = v \in [\![0, k]\!]$, corresponds to a maximal connected region $X \subseteq \Omega$ such that there exist (at least) v segmentation results $S_i \in \Omega$ that satisfy $X \subseteq S_i$. In particular, the node $(\Omega, (0)_{i=1}^k)$ is the root of $\mathcal{G}$.

This component-graph of S is much richer than the component-tree of F. Indeed, it carries not only information related to the ℓ_1 norm of the k-uples, but also additional information on their composition with respect to the k segmentation results S_i.

We build a second-layer component-tree on $\mathcal{G}$, for actually handling it. To this end, we need a valuation function $\mathcal{A}$ on the nodes $K = (X, (v_i)_{i=1}^k)$ of Θ. A relevant way consists of considering the mean gradient value along the border of X. Indeed, the higher this value, the most probable X is a significant standalone region within the image I. This valuation $\mathcal{A} : \Theta \to \mathbb{R}$ is a mixed spectral/geometric attribute on Θ, as it depends on the support X of the nodes, but also on the value of the border points of X in I.

The second-layer to be built is a max-tree, and the nodes to be preserved are located on the distal part of the tree. In other words, we build a tree $\mathcal{CT}_{max}^{root}$. The pruning can then be carried out by a simple thresholding process along the branches of the tree. Finally, since our purpose is to carry out segmentation, reconstruction policies are trivial here, and only consist of the union of the node supports.

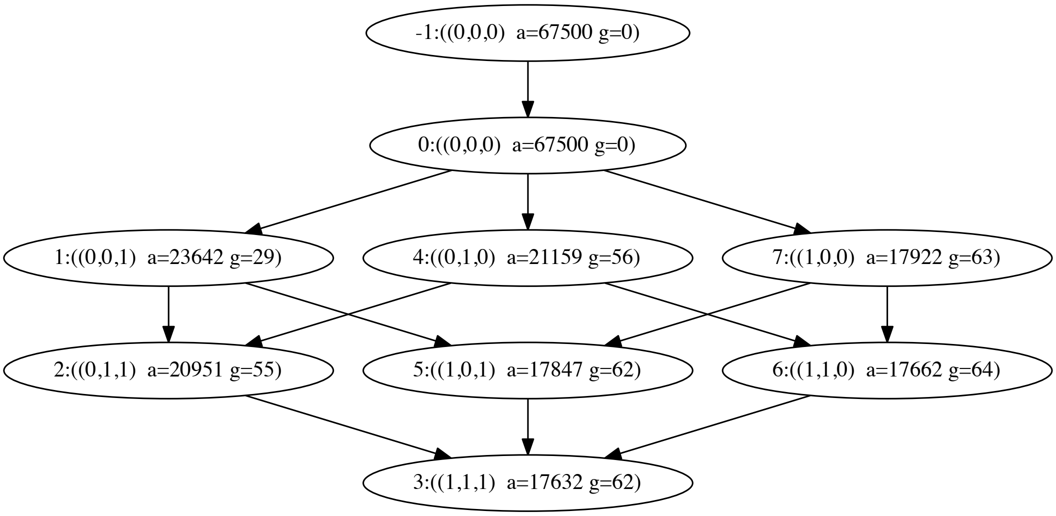


Figure 7: Component-graph $\mathcal{G}$ of the segmentation fusion image S. Each node represents a connected component of intersecting segmentations. In each node, the first number represents the node identifier; then the attributes are: the value of the node, the area of the node and the mean gradient value of the node.

To illustrate this strategy, we consider an image extracted from the segmentation evaluation database[3] [72] consisting of grey-level (natural) images along with their manual delineations (ground-truths) from three humans. The chosen example image is provided in Fig. 6. The three manual segmentations are fused using the function S given in Eq. (21). Then, each point of S has a value $v \in \{0,1\}^3 = \mathbb{V}$. Since $\mathbb{V}$ is endowed with the (partial) component-wise ordering $\leq$, extremal value of S are $(0,0,0)$ (the point does not belong to any segmentation) and $(1,1,1)$ (the point belongs to all segmentations).

The component-graph $\mathcal{G}$ constructed from the image S is represented in Fig. 7. From $\mathcal{G}$, we build the component-tree (max-tree) $\mathcal{CT}$ (Fig. 8). This component-tree is based on the successive thresholdings of the mean gradient attribute. The leaves represent the set of extremal regions of $\mathcal{G}$ according to the mean gradient criterion. For instance, in Fig. 8, the leaf node at level $h = 64$ represents the isolated node of $\mathcal{G}$ at level $v = (1,1,0)$. Therefore, this node represents the connected component with the highest mean gradient on its contour, resulting from the intersection between the manual segmentations (see Fig. 6(b,c)). It can be interpreted as the optimal fusion of segmentations according to the chosen criterion. This result is illustrated by Fig. 9.

7 Conclusion

By coupling the two recently introduced notions of shaping and component-graph, we opened 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 such an approach, in the field of multivalued mathematical morphology. Beyond encouraging results obtained on application examples, this work raises various algorithmic, methodological and applicative perspectives.

[3] http://www.wisdom.weizmann.ac.il/~vision/Seg_Evaluation_DB/index.html

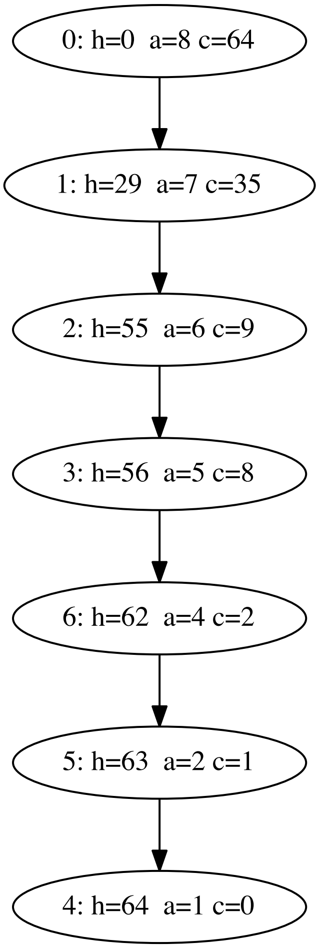


Figure 8: Component-tree on the component-graph $\mathcal{G}$ of Fig. 7. Each node at level *h* represents a connected component of $\mathcal{G}$ obtained by thresholding the mean gradient attribute *g* in $\mathcal{G}$ at value *h*. Attributes *a* and *c* in the component-tree represent the area (i.e. the number of graph-nodes of Fig. 7 included in the connected component) and contrast (or height) of the node, respectively.

From an algorithmic point of view, we have only considered here the case of multiband 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 without suprema / infima 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 fulfilled 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 involve them in our proposed framework.

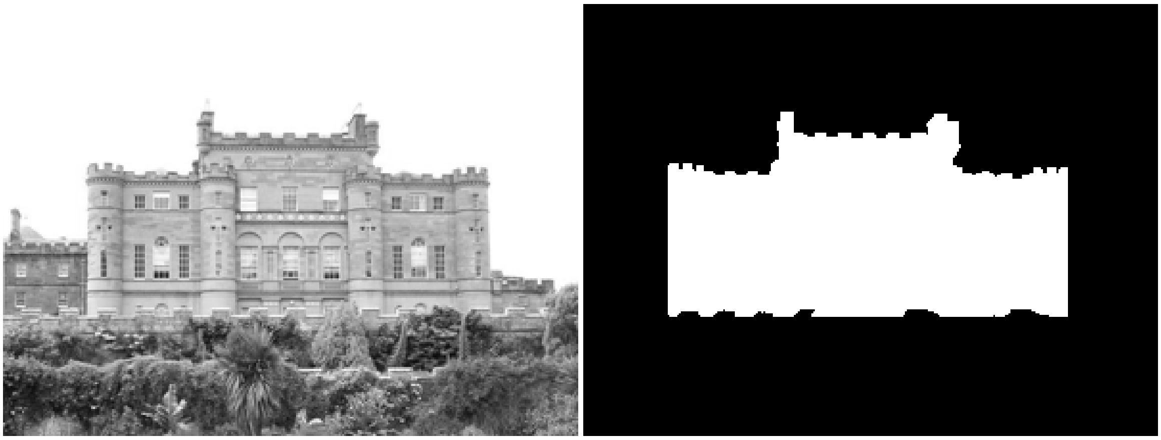


Figure 9: Left: original image. Right: best fusion of segmentations according to the given criterion (mean gradient of the contour).

From a methodological point of view, we have considered scalar attributes as valuation for the inner layer of component-graph. Such scalar attributes are generally defined on integers or real numbers, and then equipped with a canonical total order. The outer layer of the proposed data structure is therefore a component-tree. More generally, the explicit handling of vectorial attributes [70, 71] at the inner layer of the structure would lead to the construction of a component-graph at the outer 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 [62, 63].

From an applicative point of view, the proposed framework has already been used 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) [17, 65]. 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 involve combining different modalities, which is often complicated. In this context, coupling component-trees / graphs with other kinds of hierarchies (trees of shapes, binary partition trees, or watershed hierarchies) may constitute an promising perspective.

Resources 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.

Acknowledgement: This research was partially funded by the *Programme d'Investissements d'Avenir* (LabEx Bézout, ANR-10-LABX-58; France Life Imaging, ANR-11-INBS-0006). The authors thank Michel 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] Bosilj, P., Kijak, E., Lefèvre, S.: Partition and inclusion hierarchies of images: A comprehensive survey. Journal of Imaging **4**(2), 33 (2018)

[32] Jones, R.: Connected filtering and segmentation using component trees. Computer Vision and Image Understanding **75**(3), 215–228 (1999)

[33] 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)

[34] 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)

[35] 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)

[36] Naegel, B., Wendling, L.: A document binarization method based on connected operators. Pattern Recognition Letters **31**(11), 1251–1259 (2010)

[37] Guigues, L., Cocquerez, J.P., Le Men, H.: Scale-sets image analysis. International Journal of Computer Vision **68**(3), 289–317 (2006)

[38] 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)

[39] 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)

[40] Breen, E.J., Jones, R.: Attribute openings, thinnings, and granulometries. Computer Vision and Image Understanding **64**(3), 377–389 (1996)

[41] 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)

[42] 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)

[43] 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)

[44] Passat, N., Naegel, B.: An extension of component-trees to partial orders. In: ICIP, International Conference on Image Processing, Proceedings, pp. 3981–3984 (2009)

[45] Passat, N., Naegel, B.: Component-trees and multivalued images: Structural properties. Journal of Mathematical Imaging and Vision **49**(1), 37–50 (2014)

[46] Kurtz, C., Naegel, B., Passat, N.: Connected filtering based on multivalued component-trees. IEEE Transactions on Image Processing **23**(12), 5152–5164 (2014)

[47] Naegel, B., Passat, N.: Colour image filtering with component-graphs. In: ICPR, International Conference on Pattern Recognition, Proceedings, pp. 1621–1626 (2014)

[48] 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)

[49] Salembier, P., Serra, J.: Flat zones filtering, connected operators, and filters by reconstruction. IEEE Transactions on Image Processing **4**(8), 1153–1160 (1995)

[50] Heijmans, H.J.A.M.: Connected morphological operators for binary images. Computer Vision and Image Understanding **73**(1), 99–120 (1999)

[51] 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)

[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.: A morphological tree of shapes for color images. In: ICPR, International Conference on Pattern Recognition, Proceedings, pp. 1132–1137 (2014)

[63] Carlinet, E., Géraud, T.: MToS: A tree of shapes for multivariate images. IEEE Transactions on Image Processing **24**(12), 5330–5342 (2015)

[64] Xu, Y., Géraud, T., Najman, L.: Connected filtering on tree-based shape-spaces. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(6), 1126–1140 (2016)

[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] Passat, N., Naegel, B., Kurtz, K.: Implicit component-graph: A discussion. In: ISMM, International Symposium on Mathematical Morphology, Proceedings, *Lecture Notes in Computer Science*, vol. 10225, pp. 235–248. Springer (2017)

[67] Passat, N., Naegel, B., Kurtz, K.: Component-graph construction. Journal of Mathematical Imaging and Vision **61**(6), 798–823 (2019)

[68] Grossiord, E.: Hierarchical approaches for multi-modal image analysis. Application to lymphoma segmentation from PET/CT for lesions volume quantification. (In English.) PhD thesis, Université Paris-Est (2017)

[69] Urien, H.: Détection et segmentation de lésions dans des images cérébrales TEP-IRM. (In French.) PhD thesis, Telecom ParisTech (2018)

[70] 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)

[71] Naegel, B., Passat, N., Boch, N., Kocher, M.: Segmentation using vector-attribute filters: Methodology and application to dermatological imaging. In: ISMM, International Symposium on Mathematical Morphology, Proceedings, vol. 1, pp. 239–250. (2007)

[72] Alpert, S., Galun, M., Brandt, A., Basri, R.: Image segmentation by probabilistic bottom-up aggregation and cue integration. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(2), 315–327 (2012)