Connected filtering based on multivalued component-trees

Camille Kurtz, Benoît Naegel, Nicolas Passat

HAL Id: hal-01694358
https://hal.univ-reims.fr/hal-01694358v1

Submitted on 1 Mar 2018

Connected filtering based on component-graphs: Multivalued component-trees

Camille Kurtz, Benoît Naegel, Nicolas Passat

Abstract—In recent works, a new notion of component-graph was introduced. It extends the classical notion of component-tree initially proposed in mathematical morphology to model the structure of grey-level images. Component-graphs can indeed model the structure of any – grey-level or multivalued – images. We now extend the antiextensive filtering scheme based on component-trees, in order to make it tractable in the framework of component-graphs. More precisely, we provide solutions for building a component-graph; reducing it based on selection criteria; and reconstructing a filtered image from a reduced component-graph. In this article, we first consider the cases where component-graphs still have a tree structure; they are then called multivalued component-trees. The relevance and usefulness of such multivalued component-trees are illustrated by applicative examples on hierarchically classified remote sensing images.

Index Terms—Component-graphs, component-trees, multivalued component-trees, antiextensive filtering, connected operators, partially ordered sets, mathematical morphology.

I. INTRODUCTION

CONNECTED operators [1] gather image processing tools defined in the framework of mathematical morphology and successfully applied in a wide spectrum of applications (see [2][3, Ch. 7] for recent surveys). In this context, the notion of component-tree [4] has received a specific attention.

The component-tree is a hierarchical data structure that models some characteristics of a grey-level image by considering its binary level-sets obtained from successive thresholdings. Component-trees are particularly well-suited for the design of methods devoted to process grey-level images, based on hypotheses related to the topology (connectedness) and the specific intensity (local extrema) of structures of interest. Based on these properties, component-trees have been involved in several image processing applications, especially for filtering and segmentation.

The success of component-trees in the field of grey-level image processing, together with the increasing need for applications involving multivalued images, has motivated their extension to the case of such images, which can take their values in any – totally or partially – ordered sets. After a preliminary study of the relations between component-trees and multivalued images [5], a generalisation of component-trees to such images has been initiated in [6], leading to a new notion of *component-graph*.

Camille Kurtz is with Université Paris Descartes, LIPADE, Paris, France (camille.kurtz@parisdescartes.fr).

Benoît Naegel is with Université de Strasbourg, CNRS, ICube, Strasbourg, France (b.naegel@unistra.fr).

Nicolas Passat is with Université de Reims Champagne-Ardenne, CReSTIC, Reims, France (nicolas.passat@univ-reims.fr).

A theoretical study of the structural properties of component-graphs has been proposed in [7]. We now focus on the algorithmic key-points that lead to the effective development of filtering procedures based on this notion. More precisely, we propose an extension of the classical antiextensive filtering scheme based on component-trees [4], [8], and thus initially devoted only to grey-level images.

Our work, that contributes to the extension of mathematical morphology to multivalued images, is divided in two parts. In the present article, we deal with the case where the component-graphs preserve a tree structure, despite the partial order defined on the image values; in such case, the component-graphs are called *multivalued component-trees*. (The case where component-graphs are no longer trees will be considered in a further study, that will focus on specific algorithmic issues, that differ from those evoked hereinafter.)

This article – which is an extended and improved version of the conference paper [9] – is organised as follows. Sec. II describes the context of this work. Sec. III provides useful background notions. Sec. IV recalls the notion of component-graph. Sec. V describes the antiextensive filtering scheme initially developed for component-trees and extended to component-graphs. Sec. VI provides the conditions under which a component-graph is indeed a multivalued component-tree. Secs. VII–IX then provide algorithmic solutions to deal with the successive steps of the antiextensive filtering scheme in that case: Sec. VII describes how to build a multivalued component-tree; Sec. VIII discusses various ways to reduce it; and Sec. IX deals with the issue of image reconstruction from a reduced multivalued component-tree. Sec. X presents experimental results. Sec. XI concludes this work.

II. RELATED WORKS

A. Component-tree

Initially proposed in the field of statistics [10], [11], the component-tree (also known as dendrone [12], [13], confinement tree [14] or max-tree [4]) has been (re)defined in the framework of mathematical morphology and involved in the development of morphological operators [4], [8].

From a methodological point of view, some efforts have been conducted to enable the efficient computation of component-trees [4], [15], [16], [17] (see also [18] for a comparative study). From an applicative point of view, component-trees have been involved in the development of several image processing and analysis techniques. Most are devoted to filtering or segmentation [4], [8], [19], [20], [21]. However, other applications have also been considered, e.g., image registration

[22], [14], retrieval [23], [24], classification [25], visualisation [26], or document binarisation [27].

In the field of filtering and segmentation, many methods have been designed to detect some structures of interest by using information modelled by attributes [28], [29], and stored at each node of the tree. These attributes are chosen according to hypotheses related to the applicative context. The subtree obtained by reducing the component-tree of an image with respect to these attributes, can then be used to reconstruct a binary (segmentation) or grey-level (filtering) result.

B. Multivalued mathematical morphology

Mathematical morphology has been first defined on binary images, and then on grey-level ones [30]. Its extension to multivalued (e.g., colour, multispectral, label) images is an important task, motivated by potential applications in multiple areas. Several contributions have been devoted to this specific purpose (a whole state of the art is beyond the scope of this article; see [31] for a recent survey).

In general, the sets in which such multivalued images take their values are not canonically equipped with total orders, but with partial ones. Various strategies have been considered to deal with this issue. Except in few works (see, e.g., [32]), they intend to split these value sets into several totally ordered ones (marginal processing), or to define ad hoc total order relations on them (vectorial processing), with several variants [33], [34], [35], [36], [37], [38].

These approaches present the advantage of embedding multivalued images into simpler models which authorise to process them similarly to grey-level ones, enabling in particular to reduce the algorithmic complexity induced by partially ordered sets. However, they also potentially bias the information intrinsically carried by these – more complex but richer – partially ordered value sets.

In the present work, we deal with the general case of *any* (partially or totally) ordered value sets, without attempting to modify the order, then providing a contribution to the extension of mathematical morphology to multivalued images.

III. Definitions and notations

This section gathers definitions and notations required to make this work self-contained.

A. Notations

The inclusion (resp. strict inclusion) on sets is noted $\subseteq$ (resp. $\subset$). The cardinality of a set X is noted $|X|$. The power set of a set X is noted 2^X.

A function F from a set X to a set Y is noted $F : X \to Y$, and the set of all the functions from X to Y is noted Y^X. If $X' \subseteq X$ and $Y' \subseteq Y$, we note $F(X') = \{F(x) \mid x \in X'\}$ and $F^{-1}(Y') = \{x \in X \mid F(x) \in Y'\}$. If F is a bijection, we also note F^{-1} its inverse function.

Let $\asymp$ be a (binary) relation on a set X. The restriction of $\asymp$ to a subset $Y \subseteq X$ will generally still be noted $\asymp$ (except if a new notation is introduced).

We say that $\asymp$ is an equivalence relation if $\asymp$ is reflexive, transitive and symmetric. For any $x \in X$, the equivalence class of x with respect to $\asymp$ is noted $[x]_\asymp$. The quotient set of all these equivalence classes is noted $X/\asymp$.

We say that $\asymp$ is an order relation (and that $(X, \asymp)$ is an ordered set) if $\asymp$ is reflexive, transitive and antisymmetric. Moreover, we say that $\asymp$ is a total (resp. partial) order relation (and that $(X, \asymp)$ is a totally (resp. partially) ordered set), if $\asymp$ is total (resp. partial), i.e., if $\forall x, y \in X, x \asymp y \vee y \asymp x$ (resp. if $\exists x, y \in X, x \not\asymp y \wedge y \not\asymp x$).

For any symbol further used to denote an order relation ($\subseteq$, $\leqslant$, $\trianglelefteq$, etc.), the inverse symbol ($\supseteq$, $\geqslant$, $\trianglerighteq$, etc.) denotes the associated dual order, while the symbol without lower bar ($\subset$, $<$, $\triangleleft$, etc.) denotes the associated strict order.

The Hasse diagram of an ordered set $(X, \leqslant)$ is the couple $(X, \prec)$ where $\prec$ is the cover relation associated to $\leqslant$, defined for all $x, y \in X$ by $x \prec y$ iff $x < y$ and there is no $z \in X$ such that $x < z < y$.

If $(X, \leqslant)$ is an ordered set and $x \in X$, we note $x^\uparrow = \{y \in X \mid x \leqslant y\}$ (resp. $x^{\uparrow\star} = \{y \in X \mid x < y\}$) and $x^\downarrow = \{y \in X \mid y \leqslant x\}$ (resp. $x^{\downarrow\star} = \{y \in X \mid y < x\}$), namely the sets of the elements greater (resp. strictly greater) and lower (resp. strictly lower) than x, respectively. If $Y, Z \subseteq X$, we write, by abuse of notation, $Y \leqslant Z$ when for any $y \in Y$ and any $z \in Z$, we have $y \leqslant z$. If $Y \subseteq X$, the set of all the maximal and minimal elements of Y are noted $\bigtriangledown^\leqslant Y$ and $\bigtriangleup^\leqslant Y$, respectively. The supremum and the infimum of Y are noted (when they exist) $\bigvee^\leqslant Y$ and $\bigwedge^\leqslant Y$, respectively (we will note $\bigcup$ and $\bigcap$ for $\bigvee^\subseteq$ and $\bigwedge^\subseteq$, respectively). The maximum and the minimum of Y are noted (when they exist) $\curlyvee^\leqslant Y$ and $\curlywedge^\leqslant Y$, respectively. An ordered set $(X, \leqslant)$ is a lattice if for any $x, y \in X$, $\bigvee^\leqslant\{x, y\}$ and $\bigwedge^\leqslant\{x, y\}$ exist.

B. Basic notions

Let Ω be a nonempty finite set. Let V be a nonempty finite set equipped with an order relation $\leqslant$. We assume that $(V, \leqslant)$ admits a minimum, noted $\bot$.

An image is a function

$$
\begin{array}{rrcl}
I & : & \Omega & \longrightarrow & V \\
& & x & \longmapsto & v
\end{array} \tag{1}
$$

The sets Ω and V are called the support and the value set of I, respectively. For any $x \in \Omega$, $I(x) \in V$ is the value of I at x. Without loss of generality, we assume that $I^{-1}(\{\bot\}) \neq \emptyset$. If $(V, \leqslant)$ is a totally (resp. partially) ordered set, we say that I is a grey-level (resp. a multivalued) image. We note $\leq$ the pointwise order relation on V^Ω induced by $\leqslant$.

For any $v \in V$, the thresholding function at value v is defined by

$$
\begin{array}{rrcl}
\lambda_v & : & V^\Omega & \longrightarrow & 2^\Omega \\
& & I & \longmapsto & \{x \in \Omega \mid v \leqslant I(x)\}
\end{array} \tag{2}
$$

For any $X \subseteq \Omega$ and any $v \in V$, the cylinder function of support X and value v is defined by

$$
\begin{array}{rrcl}
C_{(X,v)} & : & \Omega & \longrightarrow & V \\
& & x & \longmapsto & \begin{cases} v & \text{if } x \in X \\ \bot & \text{otherwise} \end{cases}
\end{array} \tag{3}
$$

In order to handle component-graphs, we need to define a notion of connectedness on Ω. In this work, we consider the

classical and versatile graph-based connectedness, generally used in digital image processing [39].

Let $\frown$ be an adjacency (i.e., irreflexive, symmetric) relation on Ω. For any $X \subseteq \Omega$, we define the connectedness relation $\leftrightarrow$ as the reflexive-transitive closure of $\frown$ on X. It is an equivalence relation, and the set $X/\leftrightarrow$, that contains the connected components of X, is noted $\mathcal{C}[X]$.

IV. COMPONENT-GRAPHS

The notion of component-graph has been initially introduced in [6]. It generalises the notion of component-tree [4], previously proposed for the representation and processing of grey-level images, i.e., images taking there values in totally ordered sets. We restrict our presentation to the definitions and properties mandatory to understand the sequel of this work. A complete description of the structural properties of component-graphs can be found in [7].

Definition 1: Let $v \in V$ and $X \in \mathcal{C}[\lambda_v(I)]$. The couple (X, v) is called a *valued connected component*. We define the set Θ of all the valued connected components of I as

$$\Theta = \bigcup_{v \in V} \mathcal{C}[\lambda_v(I)] \times \{v\} \tag{4}$$

From the order relation $\leqslant$ on V, and the inclusion relation $\subseteq$ on 2^Ω, we define the order $\trianglelefteq$ on Θ as

$$(X_1, v_1) \trianglelefteq (X_2, v_2) \Longleftrightarrow \begin{cases} (X_1 \subset X_2) \vee \\ (X_1 = X_2 \wedge v_2 \leqslant v_1) \end{cases} \tag{5}$$

In first approach, the component-graph $\mathfrak{G}$ of I is the Hasse diagram of the ordered set $(\Theta, \trianglelefteq)$. However, three variants of component-graphs can relevantly be considered by defining two other subsets $\ddot{\Theta} \subseteq \dot{\Theta} \subseteq \Theta$ of valued connected components

$$\dot{\Theta} = \{(X, v) \in \Theta \mid \forall (X, v') \in \Theta, v \not< v'\} \tag{6}$$
$$\ddot{\Theta} = \{(X, v) \in \Theta \mid \exists x \in X, v = I(x)\} \tag{7}$$

We note $\blacktriangleleft$ (resp. $\dot{\blacktriangleleft}$, resp. $\ddot{\blacktriangleleft}$) the cover relation associated to the order relation $\trianglelefteq$ on Θ (resp. to the restriction of $\trianglelefteq$ to $\dot{\Theta}$, resp. to the restriction of $\trianglelefteq$ to $\ddot{\Theta}$). We then have the following definition for the three variants of component-graphs (see Fig. 1).

Definition 2 ([7]): The Θ (resp. $\dot{\Theta}$, resp. $\ddot{\Theta}$)-*component-graph* of I is the Hasse diagram $\mathfrak{G} = (\Theta, \blacktriangleleft)$ (resp. $\dot{\mathfrak{G}} = (\dot{\Theta}, \dot{\blacktriangleleft})$, resp. $\ddot{\mathfrak{G}} = (\ddot{\Theta}, \ddot{\blacktriangleleft})$) of the ordered set $(\Theta, \trianglelefteq)$ (resp. $(\dot{\Theta}, \trianglelefteq)$, resp. $(\ddot{\Theta}, \trianglelefteq)$). The term $\mathring{\Theta}$-*component-graph* and the notation $\mathring{\mathfrak{G}} = (\mathring{\Theta}, \mathring{\blacktriangleleft})$ will sometimes be used for the three kinds of component-graphs. The elements of $\mathring{\Theta}$ are called *nodes*; the elements of $\mathring{\blacktriangleleft}$ are called *edges*; $(\Omega, \perp)$ is called the *root*; the elements of $\bigwedge^{\trianglelefteq} \mathring{\Theta}$ are called the *leaves* of the $\mathring{\Theta}$-component-graph.

The component-graph is a relevant extension of the component-tree since both notions are compatible for grey-level images.

Proposition 3 ([7]): If $(V, \leqslant)$ is a totally ordered set, then two of the three variants of component-graphs, namely $\dot{\mathfrak{G}}$ and $\ddot{\mathfrak{G}}$, are isomorphic to the component-tree.
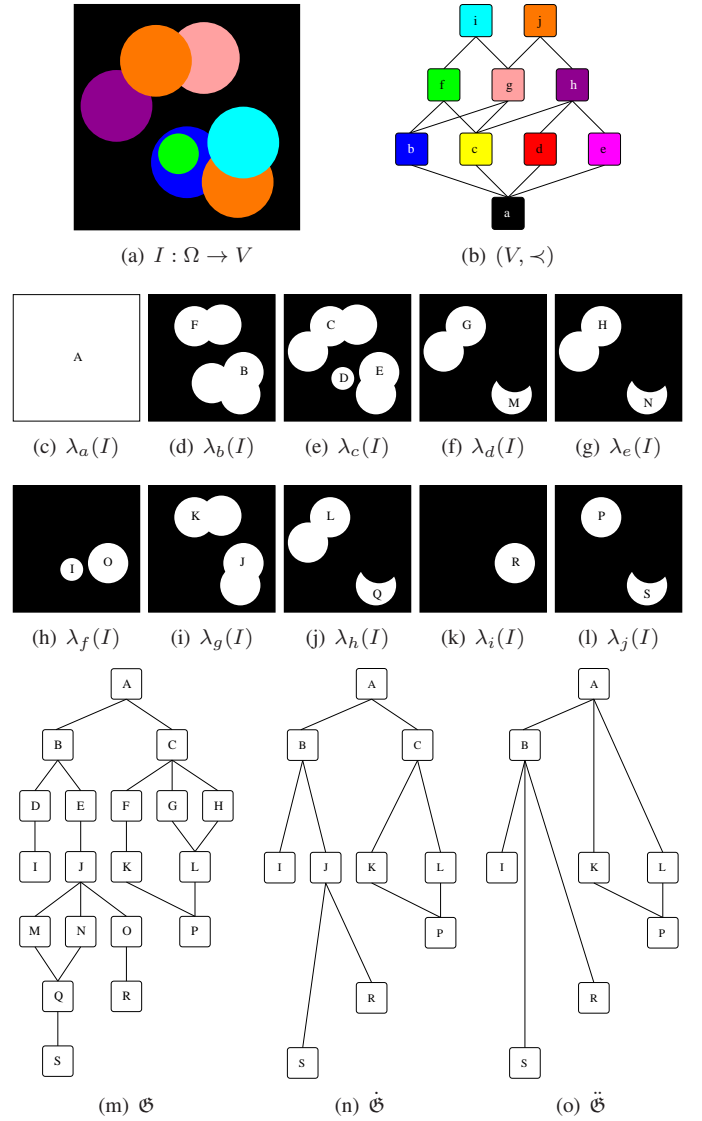

Fig. 1. (a) An image $I : \Omega \to V$, with $V = \{a, b, c, d, e, f, g, h, i, j\}$. (b) The Hasse diagram of the ordered set $(V, \leqslant)$. For the sake of readability, each value of V is associated to an arbitrary colour. (c–l) Thresholded images $\lambda_v(I)$ for $v \in V$. (m–o) The Θ, $\dot{\Theta}$ and $\ddot{\Theta}$-component-graphs of I. The letters (A–S) in nodes correspond to the associated connected components in (c–l).

V. ANTIEXTENSIVE FILTERING SCHEME

Beyond the structural compatibility expressed in Prop. 3, the component-graphs also satisfy the image (de)composition formula classically associated to component-trees. Indeed, the image I can be represented via the cylinder functions induced by the nodes of its component-graph.

Proposition 4 ([7]): We have

$$I = \bigvee_{K \in \ddot{\Theta}}^{\leqslant} C_K = \bigvee_{K \in \dot{\Theta}}^{\leqslant} C_K \tag{8}$$

In the framework of component-trees, the analogue of this formula led to the proposal of an antiextensive filtering scheme for grey-level images [4], [8]. This scheme can be extended in the framework of component-graphs. It consists of three successive steps (see Diag. (9)):

(i) construction of the component-graph $\mathring{\mathfrak{G}}$ associated to I;

(ii) reduction of $\mathring{\mathfrak{G}}$, leading to a reduced component-graph $\widehat{\mathfrak{G}}$; and

(iii) reconstruction of a filtered image $\widehat{I} \leq I$ induced by $\widehat{\mathfrak{G}}$.

$$
\begin{array}{ccc}
I & \xrightarrow{} & \widehat{I} \leq I \\
{\scriptstyle(i)}\Big\downarrow & & \Big\uparrow{\scriptstyle(iii)} \\
\mathring{\mathfrak{G}} & \xrightarrow{(ii)} & \widehat{\mathfrak{G}}
\end{array}
\tag{9}
$$

Due to the complex structure of arbitrary images and component-graphs – compared to grey-level images and component-trees – extending this scheme is not straightforward. In particular, each step may present specific difficulties.

These difficulties depend on the structure of the component-graph, and in particular on the fact that it is a tree or not. The purpose of this work is to provide algorithmic solutions for making this filtering scheme tractable in the first case, i.e., when component-graphs are *multivalued component-trees*.

VI. MULTIVALUED COMPONENT-TREES

When considering grey-level images, i.e., when $(V, \leqslant)$ is totally ordered, the component-graph $\mathring{\mathfrak{G}}$ has a tree structure, such as the component-tree. We precise hereafter what is the exact meaning of "tree structure".

Definition 5: Let $(A, \sqsubseteq)$ be an ordered set. We say that $(A, \sqsubseteq)$ is an *upper* (resp. *lower*) *piecewise totally ordered set* (*UPTOS*) (resp. (*LPTOS*)) if for any $a \in A$, $(a^{\uparrow}, \sqsubseteq)$ (resp. $(a^{\downarrow}, \sqsubseteq)$) is totally ordered. We say that the Hasse diagram of $(A, \sqsubseteq)$ *has a tree structure* if $(A, \sqsubseteq)$ is a UPTOS or a LPTOS.

The first question that arises is: *What are necessary and sufficient conditions on $(V, \leqslant)$ for guaranteeing that the component-graph $\mathring{\mathfrak{G}}$ of any image $I : \Omega \rightarrow V$ has a tree structure?* The answer is provided by the following proposition.

Proposition 6: The component-graph $\mathring{\mathfrak{G}}$ of any image $I : \Omega \rightarrow V$ has a tree structure if and only if $(V, \leqslant)$ is a LPTOS.

Proof: Let $(V, \leqslant)$ be a LPTOS. Let $I : \Omega \rightarrow V$ and $\mathring{\mathfrak{G}}$ its component-graph. Let $K = (X, v) \in \bigwedge^{\trianglelefteq} \mathring{\Theta}$. Let $K_1 = (X_1, v_1), K_2 = (X_2, v_2) \in K^{\uparrow}$. From Eq. (5), it comes $v_1, v_2 \leqslant v$. As $(V, \leqslant)$ is a LPTOS, we can assume $v_1 \leqslant v_2$. Then, Eq. (4) implies $X_2 \subseteq X_1$, and thus $K_2 \trianglelefteq K_1$. Finally, $(K^{\uparrow}, \trianglelefteq)$ is totally ordered, and $(\mathring{\Theta}, \trianglelefteq)$ is a UPTOS.

For any $I : \Omega \rightarrow V$ we assume that $\mathring{\mathfrak{G}}$ has a tree structure. As $(\mathring{\Theta}, \trianglelefteq)$ admits a maximum, it is a UPTOS. Let $v \in \bigvee^{\leqslant} V$. Let $I : \Omega \rightarrow V$ be the constant image defined by $I(x) = v$ for all $x \in X$. Then $(\Omega \times v^{\downarrow}, \trianglelefteq)$ is totally ordered. From Eq. (4), it is isomorphic to $(v^{\downarrow}, \leqslant)$, that is then also totally ordered. Thus, $(V, \leqslant)$ is a LPTOS. ∎

In this article, we assume that $(V, \leqslant)$ is a LPTOS; in such case, the component-graphs are called *multivalued component-trees*. We study, in Secs. VII–IX, how the three steps of the antiextensive filtering scheme described in Diag. (9) can be handled in that tree case.

VII. MULTIVALUED COMPONENT-TREE CONSTRUCTION

Many efforts have been successfully devoted to efficiently build component-trees [4], [14], [15], [16], [17], [18]. The induced algorithms are consequently designed to handle grey-level images, but they cannot process images for which $(V, \leqslant)$ is a LPTOS but not a totally ordered set.

The second question that arises is then: *How can we efficiently build the multivalued component-tree of an image $I : \Omega \rightarrow V$ when $(V, \leqslant)$ is a LPTOS?*

To answer this question, we propose to "enrich" the image I. Let us first define the following two sets

$$
\begin{aligned}
C(\Omega) &= \big\{ \{x, y\} \mid x \frown y \wedge \big(I(x) \leqslant I(y) \vee I(y) \leqslant I(x) \big) \big\} \\
\overline{C}(\Omega) &= \big\{ \{x, y\} \mid x \frown y \wedge \big(I(x) \not\leqslant I(y) \wedge I(y) \not\leqslant I(x) \big) \big\}
\end{aligned}
\tag{10}
$$

that gather the adjacent points whose values in I are comparable, and non-comparable, respectively.

Let us now define the extended set

$$
\Omega_\Gamma = \Omega \cup \big\{ \varepsilon_c \big\}_{c \in \overline{C}(\Omega)}
\tag{11}
$$

that adds to Ω one new point $\varepsilon_{\{x,y\}}$ for each couple of adjacent points x, y whose values $I(x)$, $I(y)$ are not comparable.

We then enrich the image I as

$$
\left|
\begin{array}{rclcl}
I_\Gamma & : & \Omega_\Gamma & \longrightarrow & V \\
& & x \in \Omega & \longmapsto & I(x) \\
& & \varepsilon_{\{x,y\}} \in \Omega_\Gamma \setminus \Omega & \longmapsto & \bigwedge^{\leqslant} \{I(x), I(y)\}
\end{array}
\right.
\tag{12}
$$

Based on the adjacency relation $\frown$ on Ω, we define a new adjacency relation $\frown_\Gamma$ on Ω_Γ as follows

$$
\forall \{x, y\} \in C(\Omega), \ x \frown_\Gamma y
\tag{13}
$$

$$
\forall \{x, y\} \in \overline{C}(\Omega), \ \big(x \frown_\Gamma \varepsilon_{\{x,y\}} \big) \wedge \big(\varepsilon_{\{x,y\}} \frown_\Gamma y \big)
\tag{14}
$$

More generally, we extend all the notions introduced in Sec. III-B to I_Γ, by suffixing them by Γ, in case of ambiguity.

The image I and the enriched one I_Γ are strongly related. First, from Eqs. (10–14), we have, for any $v \in V$

$$
\forall X \in \mathcal{C}[\lambda_v(I)], \exists X_\Gamma \in \mathcal{C}_\Gamma[\lambda_v(I_\Gamma)], X \subseteq X_\Gamma
\tag{15}
$$

In other words, any valued connected component $K \in \Theta$ can be "embedded" in a valued connected component of $K_\Gamma \in \Theta_\Gamma$. We can then licitly define the function

$$
\left|
\begin{array}{rclcl}
\Gamma & : & \Theta & \longrightarrow & \Theta_\Gamma \\
& & (X, v) & \longmapsto & (X_\Gamma, v) \text{ such that } X \subseteq X_\Gamma
\end{array}
\right.
\tag{16}
$$

The injectivity and surjectivity of Γ follow from Eqs. (10–16). The inverse function $\Gamma^{-1} : \Theta_\Gamma \rightarrow \Theta$ of Γ is then defined, for any $(X_\Gamma, v) \in \Theta_\Gamma$, as

$$
\Gamma^{-1}((X_\Gamma, v)) = (X_\Gamma \cap \Omega, v)
\tag{17}
$$

Proposition 7: The bijection Γ induces an isomorphism between $(\Theta, \trianglelefteq)$ and $(\Theta_\Gamma, \trianglelefteq_\Gamma)$.

Proof: Let $K = (X, v), K' = (X', v') \in \Theta$. Let $K_\Gamma = \Gamma(K) = (X_\Gamma, v), K'_\Gamma = \Gamma(K') = (X'_\Gamma, v') \in \Theta_\Gamma$. From the bijectivity of Γ and Eq. (15) we have $K_\Gamma \trianglelefteq K'_\Gamma \Rightarrow K \trianglelefteq K'$. Let us now suppose that $K \trianglelefteq K'$. As $(\Theta, \trianglelefteq)$ is a UPTOS, Eq. (5) implies $v' \leqslant v$, and $X \subseteq X'$. Let $x_b \in X_\Gamma$. If $x_b \in \Omega$, then $x_b \in X \subseteq X' \subseteq X'_\Gamma$. If $x_b \notin \Omega$, then, from Eq. (10), there exist $x'_b, x''_b \in X \subseteq X'$ such that $x'_b \frown_\Gamma x_b \frown_\Gamma x''_b$ and $v \leqslant I(x'_b), I(x''_b)$. From Eq. (12), we have $v' \leqslant I(x_b)$, and then $x_b \in X'_\Gamma$. Finally, it comes $K_\Gamma \trianglelefteq K'_\Gamma$. ∎

The following result derives from this proposition and Eqs. (4,6).

Corollary 8: The multivalued component-trees $\mathfrak{G}$ and $\dot{\mathfrak{G}}$ are isomorphic to $\mathfrak{G}_\Gamma$ and $\dot{\mathfrak{G}}_\Gamma$, respectively.

Remark 9: Such an isomorphism does not hold between $\mathfrak{G}$ and $\ddot{\mathfrak{G}}_\Gamma$, due to the potential addition of nodes in $\ddot{\Theta}_\Gamma$, induced by the addition of points in Ω_Γ.

Despite this isomorphism, I_Γ is still not a grey-level image. However, it now presents an important property, by comparison with I.

Property 10: Let $x, y \in \Omega_\Gamma$. We have

$$\left(x \frown_\Gamma y\right) \Longrightarrow \left(I_\Gamma(x) \leqslant I_\Gamma(y) \vee I_\Gamma(y) \leqslant I_\Gamma(x)\right) \quad (18)$$

In other words, any adjacent points have comparable values.

Let us now consider the function

$$\left| \begin{array}{cccc} \psi & : & V & \longrightarrow & \mathbb{N} \\ & & v & \longmapsto & |v^\downarrow| \end{array} \right. \quad (19)$$

that associates to each value $v \in V$ the length of the shortest path between v and $\bot$ in $(V, \leqslant)$. The following property derives from Eq. (19).

Property 11: The function ψ induces a homomorphism from $(V, \leqslant)$ to the totally ordered set $(\mathbb{N}, \leqslant_\mathbb{N})$.

If $(V, \leqslant)$ is not totally ordered, it is plain that ψ is not injective. In other words, we may have $\psi(v) \leqslant_\mathbb{N} \psi(v')$ while $v \not\leqslant v'$. However, this does not constitute a problem when considering images $I_\Gamma : \Omega_\Gamma \to V$. Indeed, let us consider the grey-level image $I_\Psi = \psi \circ I_\Gamma$ defined as

$$\left| \begin{array}{cccc} I_\Psi & : & \Omega_\Gamma & \longrightarrow & \mathbb{N} \\ & & x & \longmapsto & \psi(I_\Gamma(x)) \end{array} \right. \quad (20)$$

We extend all the notions introduced in Sec. III-B to I_Ψ, by suffixing them by Ψ, in case of ambiguity.

From Eqs. (18–20), we have, for any $v \in V$

$$\forall X_\Gamma \in \mathcal{C}_\Gamma[\lambda_v(I_\Gamma)], \exists X_\Psi \in \mathcal{C}_\Psi[\lambda_{\psi(v)}(I_\Psi)], X_\Gamma = X_\Psi \quad (21)$$

In particular, any valued connected component $K_\Gamma \in \dot{\Theta}_\Gamma$ can be put in correspondence with a valued connected component of $K_\Psi \in \dot{\Theta}_\Psi$. We can then licitly define the function

$$\left| \begin{array}{cccc} \Psi & : & \dot{\Theta}_\Gamma & \longrightarrow & \dot{\Theta}_\Psi \\ & & (X_\Gamma, v) & \longmapsto & (X_\Gamma, v_\Psi) \text{ where } v_\Psi = \psi(v) \end{array} \right. \quad (22)$$

The injectivity and surjectivity of Ψ follow from Eqs. (19–22) and the fact that $(V, \leqslant)$ is a LPTOS. The inverse function $\Psi^{-1} : \dot{\Theta}_\Psi \to \dot{\Theta}_\Gamma$ of Ψ is defined, for any $(X_\Gamma, v_\Psi) \in \dot{\Theta}_\Psi$, as

$$\Psi^{-1}((X_\Gamma, v_\Psi)) = (X_\Gamma, \overset{\leqslant}{\bigwedge} I(X_\Gamma)) \quad (23)$$

Proposition 12: The bijection Ψ induces an isomorphism between $(\dot{\Theta}_\Gamma, \trianglelefteq_\Gamma)$ and $(\dot{\Theta}_\Psi, \trianglelefteq_\Psi)$.

Proof: Let $K_\Gamma \in \bigwedge^{\trianglelefteq_\Gamma} \dot{\Theta}_\Gamma$, and $K_\Psi = \Psi(K_\Gamma) \in \bigwedge^{\trianglelefteq_\Psi} \dot{\Theta}_\Psi$. From Eqs. (19,20), $(K_\Gamma^\uparrow, \trianglelefteq_\Gamma)$ and $(K_\Psi^\uparrow, \trianglelefteq_\Psi)$ are isomorphic. It follows that $(\dot{\Theta}_\Gamma, \trianglelefteq_\Gamma)$ and $(\dot{\Theta}_\Psi, \trianglelefteq_\Psi)$ are also isomorphic. ∎

The following result derives from this proposition and Eqs. (4,6,7).

Corollary 13: The multivalued component-tree $\dot{\mathfrak{G}}_\Gamma$ is isomorphic to $\dot{\mathfrak{G}}_\Psi$.

Algorithm 1: Construction of $\dot{\mathfrak{G}}$ when $(V, \leqslant)$ is a LPTOS

> **Input**: $I : \Omega \to V$
> **Input**: $\frown \subseteq \Omega \times \Omega$
> **Output**: $\dot{\mathfrak{G}} = (\dot{\Theta}, \blacktriangleleft)$
> 1 $\Omega_\Gamma \leftarrow \Omega$
> 2 $\frown_\Gamma \leftarrow \frown$
> 3 **foreach** $x \frown y$ **do**
> 4 $\quad$ **if** $I(x) \not\leqslant I(y)$ **and** $I(y) \not\leqslant I(x)$ **then**
> 5 $\quad\quad$ $\Omega_\Gamma \leftarrow \Omega_\Gamma \cup \{\varepsilon_{\{x,y\}}\}$
> 6 $\quad\quad$ $\frown_\Gamma \leftarrow \frown_\Gamma \setminus \{(x,y)\} \cup \{(x, \varepsilon_{\{x,y\}}), (\varepsilon_{\{x,x'\}}, y)\}$
> 7 **foreach** $x \in \Omega$ **do**
> 8 $\quad$ $I_\Psi(x) \leftarrow \psi(I(x))$
> 9 **foreach** $\varepsilon_{\{x,y\}} \in \Omega_\Gamma \setminus \Omega$ **do**
> 10 $\quad$ $I_\Psi(\varepsilon_{\{x,y\}}) \leftarrow \psi(\bigwedge^\leqslant \{I(x), I(y)\})$
> 11 **compute** $\ddot{\mathfrak{G}}_\Psi = (\dot{\Theta}_\Psi, \blacktriangleleft_\Psi)$ **for** $I_\Psi : \Omega_\Gamma \to \mathbb{N}$
> 12 $\dot{\Theta} \leftarrow \emptyset$
> 13 **foreach** $K_\Psi \in \dot{\Theta}_\Gamma$ **do**
> 14 $\quad$ $\dot{\Theta} \leftarrow \dot{\Theta} \cup \{\Gamma^{-1}(\Psi^{-1}(K_\Psi))\}$
> 15 $\blacktriangleleft \leftarrow \emptyset$
> 16 **foreach** $K_\Psi \blacktriangleleft_\Psi K'_\Psi$ **do**
> 17 $\quad$ $\blacktriangleleft \leftarrow \blacktriangleleft \cup \{(\Gamma^{-1}(\Psi^{-1}(K_\Psi)), \Gamma^{-1}(\Psi^{-1}(K'_\Psi)))\}$

Algorithm 2: Construction of $\mathfrak{G}$ when $(V, \leqslant)$ is a LPTOS

> **Input**: $\dot{\mathfrak{G}} = (\dot{\Theta}, \blacktriangleleft)$
> **Output**: $\mathfrak{G} = (\Theta, \blacktriangleleft)$
> 1 $\Theta \leftarrow \dot{\Theta}$
> 2 $\blacktriangleleft \leftarrow \emptyset$
> 3 **foreach** $(X, v) \blacktriangleleft (X', v')$ **do**
> 4 $\quad$ **if** $v' \prec v$ **then**
> 5 $\quad\quad$ $\blacktriangleleft \leftarrow \blacktriangleleft \cup \{((X,v), (X', v'))\}$
> 6 $\quad$ **else**
> 7 $\quad\quad$ **foreach** $v' < w' < v$ **do**
> 8 $\quad\quad\quad$ $\Theta \leftarrow \Theta \cup \{(X, w')\}$
> 9 $\quad\quad$ **for** $v' \prec w < v$ **do**
> 10 $\quad\quad\quad$ $\blacktriangleleft \leftarrow \blacktriangleleft \cup \{((X, w), (X', v'))\}$
> 11 $\quad\quad\quad$ **foreach** $w \leqslant w' \prec w'' \leqslant v$ **do**
> 12 $\quad\quad\quad\quad$ $\blacktriangleleft \leftarrow \blacktriangleleft \cup \{((X, w''), (X, w'))\}$

Algorithm 3: Construction of $\ddot{\mathfrak{G}}$ when $(V, \leqslant)$ is a LPTOS

> **Input**: $I : \Omega \to V$
> **Input**: $\dot{\mathfrak{G}} = (\dot{\Theta}, \blacktriangleleft)$
> **Output**: $\ddot{\mathfrak{G}} = (\ddot{\Theta}, \ddot{\blacktriangleleft})$
> 1 $\ddot{\Theta} \leftarrow \dot{\Theta}$
> 2 $\ddot{\blacktriangleleft} \leftarrow \blacktriangleleft$
> 3 **foreach** $K = (X, v) \in \ddot{\Theta}$ **do**
> 4 $\quad$ **if** $\forall x \in X, I(x) \neq v$ **then**
> 5 $\quad\quad$ $\ddot{\Theta} \leftarrow \ddot{\Theta} \setminus \{K\}$
> 6 $\quad\quad$ **for** $K' \blacktriangleright K$ **do**
> 7 $\quad\quad\quad$ $\ddot{\blacktriangleleft} \leftarrow \ddot{\blacktriangleleft} \setminus \{(K, K')\}$
> 8 $\quad\quad\quad$ **foreach** $K'' \ddot{\blacktriangleleft} K$ **do**
> 9 $\quad\quad\quad\quad$ $\ddot{\blacktriangleleft} \leftarrow \ddot{\blacktriangleleft} \setminus \{(K'', K)\} \cup \{(K'', K')\}$

From Prop. 6 and Cors. 8, 13, we have the following result.

Theorem 14: If $(V, \leqslant)$ is a LPTOS, then the multivalued component-tree $\dot{\mathfrak{G}}$ of any image $I : \Omega \to V$ is isomorphic to the multivalued component-tree $\dot{\mathfrak{G}}_\Psi$ of the grey-level image $I_\Psi : \Omega_\Gamma \to \mathbb{N}$, i.e., its component-tree.

This justifies Alg. 1, proposed for constructing the $\dot{\Theta}$-

multivalued component-tree of $I : \Omega \to V$ when $(V, \leqslant)$ is a LPTOS. Its first part consists of adding at most $|\frown|$ new points to Ω, thus leading to a computational cost of $\mathcal{O}(|\Omega| + |\frown|)$ (lines 1–10). Then, any standard component-tree construction algorithm can be applied to build the $\dot{\Theta}$-multivalued component-tree of I (line 11). It has been proved in [15] that it could be carried out in quasi-linear time with respect to $|\Omega| + |\frown|$ (see also [18] for a comparative study of several algorithms). Finally, each node of the obtained component-tree has to be post-processed to retrieve the actual nodes of the $\dot{\Theta}$-multivalued component-tree $\mathfrak{\dot{G}}$. The global cost of this post-processing (lines 12–17) is also $\mathcal{O}(|\Omega| + |\frown|)$. Consequently, the computational cost for constructing the $\dot{\Theta}$-multivalued component-tree $\mathfrak{\dot{G}}$ of an image $I : \Omega \to V$ is of the same order as the computational cost for constructing the component-tree of a grey-level image defined on a support of size $|\Omega| + |\frown|$, i.e., quasi-linear with respect to $|\Omega| + |\frown|$.

Remark 15: Practically, we generally have $|\frown| = \mathcal{O}(|\Omega|)$. In such cases, the construction of $\mathfrak{\dot{G}}$ has a computational cost that is quasi-linear with respect to $|\Omega|$.

In order to build the Θ and $\ddot{\Theta}$-multivalued component-trees, some post-processings can then be applied on $\mathfrak{\dot{G}}$.

Let us first consider the case of $\mathfrak{G}$. For any edge $(X, v) \mathbin{\dot{\blacktriangleleft}} (X', v')$ such that $v' \not\prec v$, i.e., such that there exists a value $w' \in V$ verifying $v' < w' < v$, we remove this edge and we update $\mathfrak{\dot{G}}$ as follows: (i) we add to $\dot{\Theta}$ all the nodes (X, w') such that $v' < w' < v$; and (ii) we create the respective links between all these new successive nodes, with respect to $\prec$, plus the two links with (X, v) and (X', v'). (See [7, Prop. 12] for a formal justification of these modifications.) This process is formalised in Alg. 2. The number of added nodes and added edges is equal. The computational cost of the process is then bounded by $|\Theta \setminus \dot{\Theta}|$. In the worst case, for each point x of the image, k nodes/edges are added, where k is the length of the shortest path between $\bot$ and $I(x)$ in $(V, \leqslant)$. This justifies that the computational cost for constructing the Θ-multivalued component-tree $\mathfrak{G}$ of an image $I : \Omega \to V$, from $\mathfrak{\dot{G}}$, is $\mathcal{O}(|\Omega| \cdot \max_{v \in \nabla^\leqslant V} |v^\downarrow|)$.

Let us now consider the case of $\mathfrak{\ddot{G}}$. For any node $K = (X, v) \in \dot{\Theta}$ such that for all $x \in X$, $I(x) \neq v$, we remove K from $\dot{\Theta}$ and we update $\mathfrak{\dot{G}}$ as follows: (i) we remove the unique edge $(K, K') \in \mathbin{\dot{\blacktriangleleft}}$; and (ii) we replace any edge $(K'', K) \in \mathbin{\dot{\blacktriangleleft}}$ by (K'', K'). This process is formalised in Alg. 3. In the worst case, each edge that is linked to a leaf containing a point x is duplicated k times, where k is the length of the shortest path between $\bot$ and $I(x)$ in $(V, \leqslant)$. This justifies that the computational cost for constructing the $\ddot{\Theta}$-multivalued component-tree $\mathfrak{\ddot{G}}$ of an image $I : \Omega \to V$, from $\mathfrak{\dot{G}}$, is $\mathcal{O}(|\Omega| \cdot \max_{v \in \nabla^\leqslant V} |v^\downarrow|)$.

VIII. MULTIVALUED COMPONENT-TREE REDUCTION

Once the multivalued component-tree $\mathfrak{\dot{G}}$ has been constructed, the second step of the antiextensive filtering scheme consists of reducing $\mathfrak{\dot{G}}$. This implies in particular to choose a subset of nodes $\widehat{\Theta} \subseteq \mathring{\Theta}$. Similarly to the case of component-tree-based filtering [4], [8], this choice is based on:

- a selection criterion, i.e., a Boolean predicate $\rho : \mathring{\Theta} \to \mathbb{B}$ that indicates if the nodes satisfy a required property; and

- a reduction policy which is combined with ρ to determine which parts of the multivalued component-tree should be preserved or discarded.

If ρ models an increasing criterion, i.e., if

$$K \trianglelefteq K' \Longrightarrow \big(\rho(K) \Rightarrow \rho(K') \big) \qquad (24)$$

then the way to reduce $\mathfrak{\dot{G}}$ is straightforward (in particular, the following reduction policies are equivalent).

However, if ρ models a non-increasing criterion, i.e., if Eq. (24) does not hold, then various reduction policies can be considered. For component-trees, several classical policies have been defined, including in particular the $\min$, direct, and $\max$ ones [4]. These policies remain consistent in the case of multivalued component-trees.

Definition 16 (From [4]): Let $(V, \leqslant)$ be a LPTOS. Let $\widehat{\Theta} \subseteq \mathring{\Theta}$. Let $\rho : \mathring{\Theta} \to \mathbb{B}$ be a selection criterion. The three reduction policies $\min$, direct and $\max$ are characterised by the following axiomatic (recursive) definitions of the subsets $\widehat{\Theta}_{\min}, \widehat{\Theta}_{\mathrm{direct}}, \widehat{\Theta}_{\max} \subseteq \widehat{\Theta}$.

$$(\Omega, \bot) \in \widehat{\Theta}_{\min}, \widehat{\Theta}_{\mathrm{direct}}, \widehat{\Theta}_{\max} \qquad (25)$$

$$\rho(K) \wedge \big(K' \blacktriangleright K \wedge K' \in \widehat{\Theta}_{\min} \big) \Longrightarrow K \in \widehat{\Theta}_{\min} \qquad (26)$$

$$\rho(K) \Longrightarrow K \in \widehat{\Theta}_{\mathrm{direct}} \qquad (27)$$

$$\rho(K) \vee \big(K' \blacktriangleleft K \wedge K' \in \widehat{\Theta}_{\max} \big) \Longrightarrow K \in \widehat{\Theta}_{\max} \qquad (28)$$

Remark 17: Eq. (25) is only used to guarantee the coherence of the axiomatic definitions, since the root $(\Omega, \bot)$ has actually no influence on the reconstructed image (see Eq. (8)).

The following property on the (non-)increasingness of the different policies for the different multivalued component-trees derives from Eqs. (4,6,7) and (25–28)

Property 18: We have the following inclusion relations between the nine reduced sets of nodes of $\mathring{\Theta}$.

$$
\begin{array}{ccc}
\widehat{\Theta}_{\max}(\mathfrak{\ddot{G}}) & \xrightarrow{\subseteq} \widehat{\Theta}_{\max}(\mathfrak{\dot{G}}) \xrightarrow{\subseteq} & \widehat{\Theta}_{\max}(\mathfrak{G}) \\
\subseteq \uparrow & \subseteq \uparrow & \subseteq \uparrow \\
\widehat{\Theta}_{\mathrm{direct}}(\mathfrak{\ddot{G}}) & \xrightarrow{\subseteq} \widehat{\Theta}_{\mathrm{direct}}(\mathfrak{\dot{G}}) \xrightarrow{\subseteq} & \widehat{\Theta}_{\mathrm{direct}}(\mathfrak{G}) \\
\subseteq \uparrow & \subseteq \uparrow & \subseteq \uparrow \\
\widehat{\Theta}_{\min}(\mathfrak{\ddot{G}}) & \widehat{\Theta}_{\min}(\mathfrak{\dot{G}}) & \widehat{\Theta}_{\min}(\mathfrak{G})
\end{array} \qquad (29)
$$

The process of node selection based on these policies is algorithmically linked to the structure of the component-tree, and not to the nature of the order $\leqslant$. As a consequence, in the case where $(V, \leqslant)$ is a LPTOS, the same process can be applied for selecting nodes in multivalued component-trees. In particular it is carried out by scanning $\mathfrak{\dot{G}}$ only once in a bottom-up (direct, $\max$) or in a top-down fashion ($\min$, direct). Its computational cost is then $\mathcal{O}(|\mathring{\Theta}|)$.

Remark 19: There exist various selection criteria $\rho : \mathring{\Theta} \to \mathbb{B}$. For those which are only sensitive to the spatial part X of the nodes $K = (X, v)$, the reductions of $\mathfrak{G}$ and $\mathfrak{\dot{G}}$ actually lead to the same result. Consequently, considering the – richer – multivalued component-tree $\mathfrak{G}$ instead of $\mathfrak{\dot{G}}$ makes sense only if the chosen selection criterion ρ takes into account the values v of the nodes or, more generally, the structure of the multivalued component-tree.

IX. Filtered image reconstruction

Let $\widehat{\Theta} \subseteq \mathring{\Theta}$ be the subset of nodes in $\widehat{\mathfrak{G}}$ obtained after the reduction of $\mathring{\mathfrak{G}}$. The last step now consists of reconstructing the filtered image $\widehat{I} : \Omega \to V$.

When $(V, \leqslant)$ is a LPTOS, for any $x \in \Omega$, $(I(x)^{\downarrow}, \leqslant)$ is totally ordered, and so is $(\{C_K(x) \mid K \in \widehat{\Theta}\}, \leqslant)$. This justifies the following result.

Property 20: If $(V, \leqslant)$ is a LPTOS, then $\widehat{I}$ can be recovered by substituting $\widehat{\Theta}$ to $\mathring{\Theta}$ in Eq. (8).

In other words, when $(V, \leqslant)$ is a LPTOS, the reconstruction of the filtered image $\widehat{I}$ does not present any difficulties, and can in particular be carried out in the same way as for component-trees. Its actual computation is performed by scanning $\mathring{\mathfrak{G}}$ in a bottom-up fashion in order to update the value of each point $x \in \Omega$. The induced computational cost is then $\Omega(|\mathring{\Theta}| + |\Omega|)$.

Remark 21: Practically, it is possible to handle an image $I : \Omega \to V$ by considering its flat zones [1], i.e., the maximal connected components $X \subseteq \Omega$ such that $I(X) = \{v\}$ for given values $v \in V$, instead of its points. Due to the intrinsic properties of connectedness, the flat zones can be relevantly substituted to the points of Ω for defining the image I and its structure. In particular, the multivalued component-tree of the image I and its "flat zone analogue" image are isomorphic, and the antiextensive filtering scheme then leads to the same results for both. The preprocessing that consists of computing the set Φ of the flat zones of I, the associated image $I_\Phi : \Phi \to V$, and the adjacency $\frown_\Phi$ on Φ induced by $\frown$, presents a computational cost of $\mathcal{O}(|\Omega| + |\frown|)$. Afterwards, it is possible to substitute the set of flat zones Φ to Ω in the above complexity analyses for the three steps, until the final reconstruction of $\widehat{I}$ from $\widehat{I}_\Phi$.

As any connected operator, the multivalued component-tree-based antiextensive filtering does not introduce new contours in the filtered images. However, by contrast with the case of grey-level image filtering based on component-trees, it may happen that new values appear in the filtered images (see, e.g., Fig. 2).

Property 22: Let $(V, \leqslant)$ be a LPTOS. Let $I : \Omega \to V$ be an image, and $\mathring{\mathfrak{G}}$ its multivalued component-tree. Let $\widehat{I}$ be a filtered image obtained by antiextensive filtering of I, based on $\mathring{\mathfrak{G}}$. If $\mathring{\mathfrak{G}} = \ddot{\mathfrak{G}}$, then we have $\widehat{I}(\Omega) \subseteq I(\Omega)$. However, if $\mathring{\mathfrak{G}} = \mathfrak{G}$ or $\dot{\mathfrak{G}}$ we may have $\widehat{I}(\Omega) \not\subseteq I(\Omega)$.

In other words, only the $\ddot{\Theta}$-multivalued component-trees (and actually the isomorphic $\dot{\Theta}$-multivalued component-trees if $(V, \leqslant)$ is totally ordered) guarantee to preserve the values of the initial image.

X. Experiments

A. Influence of the choice of multivalued component-tree and reduction policy

Before considering a real application, in Section X-B, let us illustrate the effects of the choice of the multivalued component-tree and the reduction policy, when filtering a multivalued image.

To this end, we consider the synthetic example of Fig. 2, that provides a multivalued image $I : \Omega \to V$ (Fig. 2(a)) taking its values in a LPTOS $(V, \leqslant)$ (Fig. 2(b)). This image is associated
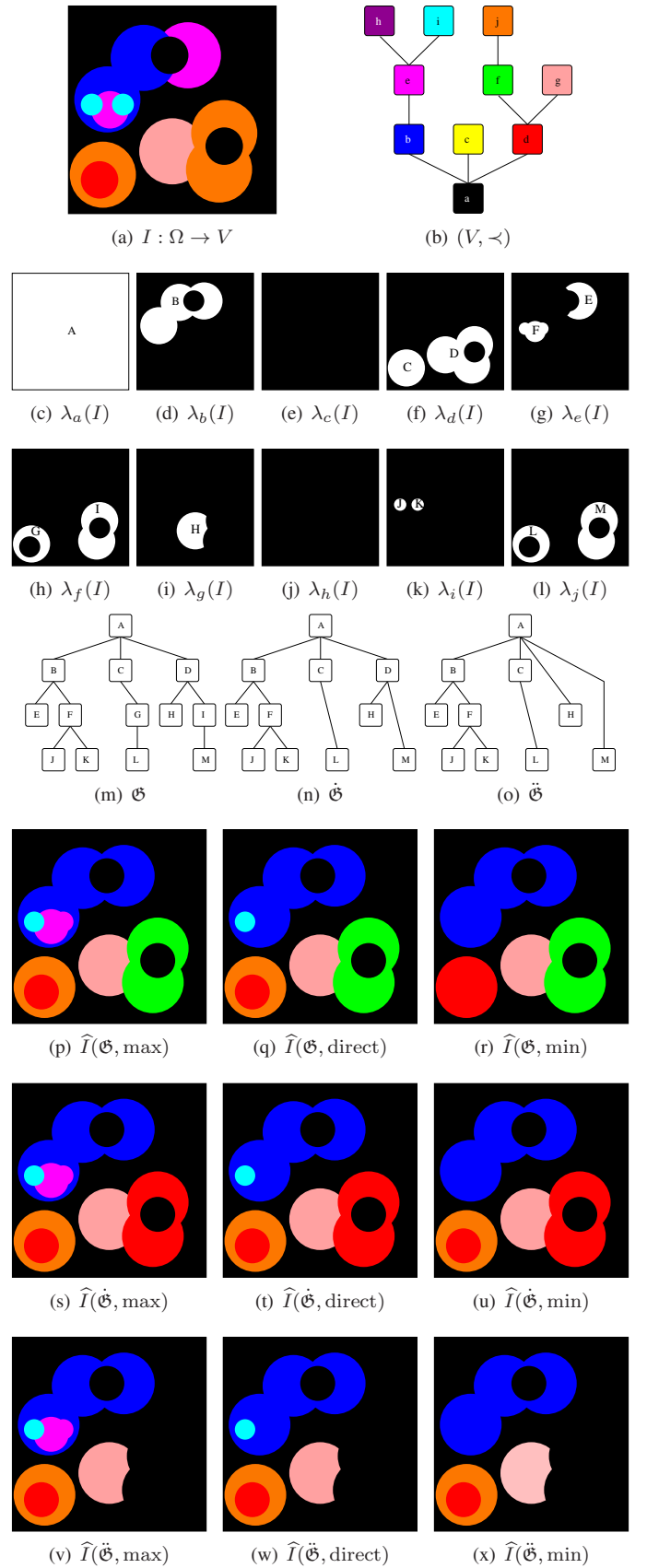

(a) $I : \Omega \to V$ 　　　　(b) $(V, \prec)$

(c) $\lambda_a(I)$ 　(d) $\lambda_b(I)$ 　(e) $\lambda_c(I)$ 　(f) $\lambda_d(I)$ 　(g) $\lambda_e(I)$

(h) $\lambda_f(I)$ 　(i) $\lambda_g(I)$ 　(j) $\lambda_h(I)$ 　(k) $\lambda_i(I)$ 　(l) $\lambda_j(I)$

(m) $\mathfrak{G}$ 　　　　(n) $\dot{\mathfrak{G}}$ 　　　　(o) $\ddot{\mathfrak{G}}$

(p) $\widehat{I}(\mathfrak{G}, \max)$ 　(q) $\widehat{I}(\mathfrak{G}, \text{direct})$ 　(r) $\widehat{I}(\mathfrak{G}, \min)$

(s) $\widehat{I}(\dot{\mathfrak{G}}, \max)$ 　(t) $\widehat{I}(\dot{\mathfrak{G}}, \text{direct})$ 　(u) $\widehat{I}(\dot{\mathfrak{G}}, \min)$

(v) $\widehat{I}(\ddot{\mathfrak{G}}, \max)$ 　(w) $\widehat{I}(\ddot{\mathfrak{G}}, \text{direct})$ 　(x) $\widehat{I}(\ddot{\mathfrak{G}}, \min)$

Fig. 2. (a) An image $I : \Omega \to V$ with $V = \{a, b, c, d, e, f, g, h, i, j\}$. (b) The Hasse diagram of the LPTOS $(V, \leqslant)$. (c–l) Thresholded images $\lambda_v(I)$ for $v \in V$. (m–o) The multivalued component-trees of I. (p–x) Filtered images $\widehat{I}$, depending on the chosen multivalued component-tree and reduction policy (see text).
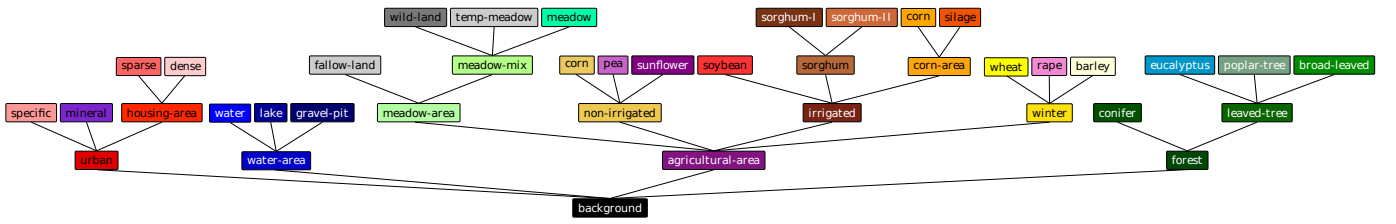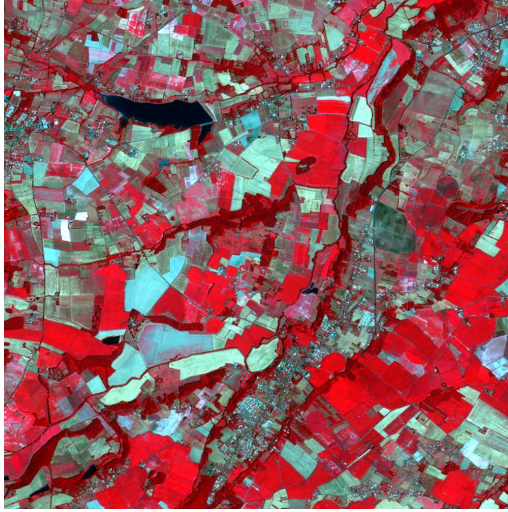
Fig. 3.   Hasse diagram of a LPTOS $(V, \leqslant)$ where V contains 40 semantic labels, and $\leqslant$ provides the generalisation / specialisation links between them.

(a) Satellite image $S : \Omega \to \mathbb{R}^4$

(b) Classification map $I^c : \Omega \to V$

Fig. 4.   (a) Satellite image (FORMOSAT-2), on a support Ω of 1000×1000 pixels. Spatial resolution: 8 m $\times$ 8 m. Spectral resolution: 4 bands (near infrared, red, green, blue). (b) Classification map $I^c : \Omega \to V$ obtained from (a) (courtesy J. Inglada, D. Ducrot, C. Marais-Sicre, O. Hagolle and M. Huc, CESBIO).


(a) $\tau = 10\,000$ m^2

(b) $\tau = 50\,000$ m^2

(c) $\tau = 100\,000$ m^2

(d) $\tau = 500\,000$ m^2

(e) $[0, 10\,000]$ m^2

(f) $[10\,000, 50\,000]$ m^2

(g) $[50\,000, 100\,000]$ m^2

(h) $[100\,000, 200\,000]$ m^2
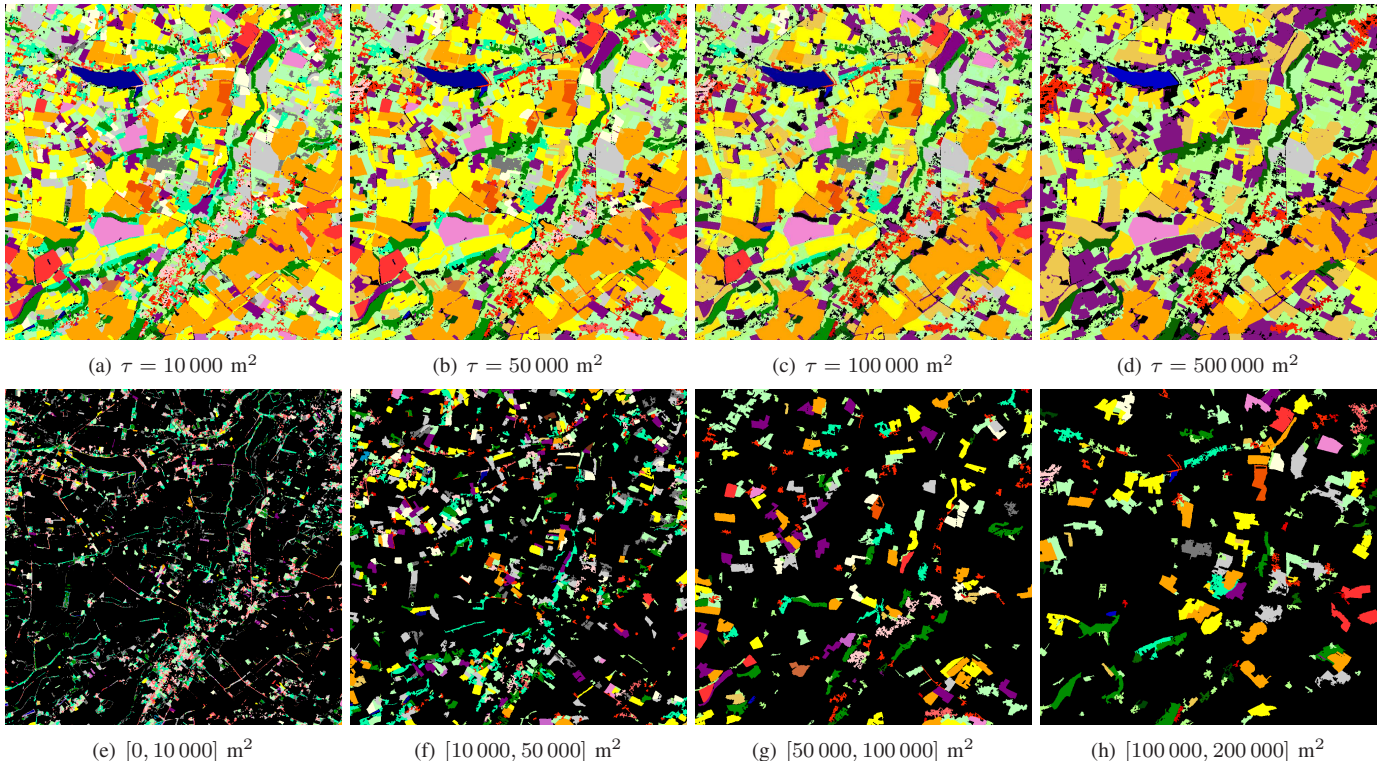
Fig. 5.   Antiextensive filtering of I^c (Fig. 4(b)), based on an area attribute $\mathcal{A}$. While I^c takes its values within the 26 leaves of $(V, \prec)$ (Fig. 3), the filtered images can also contain other values, then modelling areas with less specific semantics. (a–d) Results for $\tau < \mathcal{A}$. (e–h) Results for $\mathcal{A} \in [\tau_{min}, \tau_{max}]$.
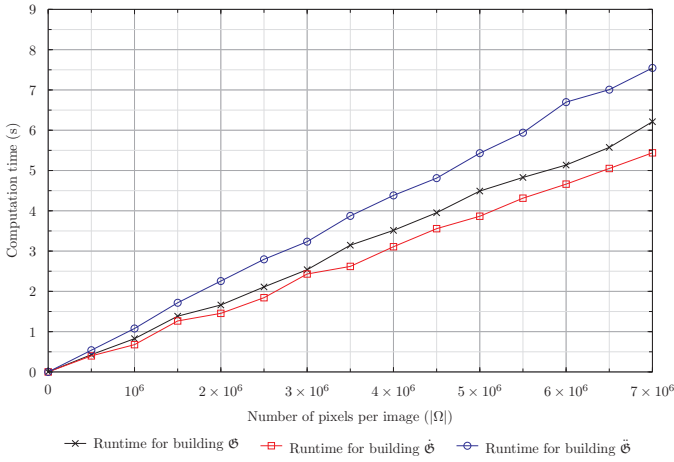
(a) Computation time required for building the multivalued component-trees.
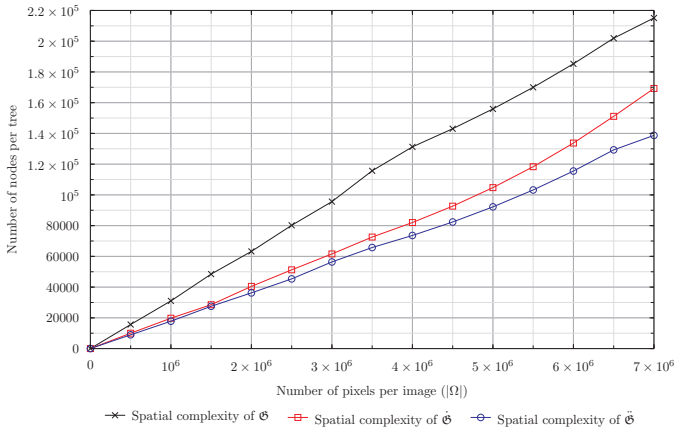
(b) Number of nodes composing the different multivalued component-trees.

Fig. 6. Computational and spatial complexities related to the construction of the multivalued component-trees of an image $I^c : \Omega \to V$ by varying the size ($|\Omega|$) of the image.

to three distinct multivalued component-trees $\mathfrak{G} = (\Theta, \blacktriangleleft)$, $\dot{\mathfrak{G}} = (\dot{\Theta}, \blacktriangleleft)$, $\ddot{\mathfrak{G}} = (\ddot{\Theta}, \ddot{\blacktriangleleft})$ (Fig. 2(m–o)), with $\Theta = \{A, B, \ldots, M\}$ (Fig. 2(c–l)).

Let us now suppose that for a given selection criterion $\rho : \Theta \to \mathbb{B}$, we have $\rho^{-1}(\{\text{true}\}) = \{A, B, C, D, H, I, J, L\}$. By reducing the three multivalued component-trees $\ddot{\mathfrak{G}}$, with the three policies max, direct, min, we obtain nine reduced multivalued component-trees from which we can reconstruct nine filtered images (Fig. 2(p–x)). Their respective properties (recovery of initially non-selected nodes; removal of initially selected nodes; appearance of new values; . . .) provide a wide range of possibilities, that may fit various kinds of use cases.

B. Application example: Filtering of hierarchical classification maps

A classical application field that involves images taking their values in ordered sets $(V, \leqslant)$ that are LPTOS, is the processing of hierarchical classification maps [40]. Such hierarchical maps, which can be obtained from supervised or unsupervised classification, are widely considered in the context of remote sensing, for instance when analysing multiresolution [41] or multitemporal satellite images [42].

In such cases, $(V, \leqslant)$ is a hierarchy of concepts. In other words, the set V is composed of semantic labels, while the relation $\leqslant$ provides some generalisation / specialisation links between them. We then obtain a tree structure $(V, \prec)$, in which the leaves of $\bigvee^{\leqslant} V$ are the most precise labels – that correspond to the smallest details of related images – while the root $\bigwedge^{\leqslant} V$ is the most general label – that is valid within the whole images.

In the proposed application example, we define a set V of labels that are related to the semantic elements of agricultural areas. These labels are illustrated in Fig. 3. They are organised from the root (background) to 26 leaves that are gathered into wide families (urban, water, agricultural, forest), progressively refined into 3 to 5 layers.

We then consider the satellite image illustrated in Fig. 4(a), that represents a complex agricultural zone. This image, defined on a set Ω of 1000×1000 pixels with a spatial resolution of 64 m^2 per pixel, has then been classified into 26 classes, corresponding to the leaves of $(V, \leqslant)$, thus leading to the classification map $I^c : \Omega \to V$ depicted in Fig. 4(b). This classification task is indeed challenging since the latest generation of remote sensing images presents high spectral and spatial resolution properties, leading to huge volumes of data. This motivates in particular denoising procedures on such kinds of images.

Based on the multivalued component-tree of $I^c : \Omega \to V$, two series of antiextensive filterings have been computed. The first has been performed by considering an increasing criterion, related to the minimal values of an area attribute $\mathcal{A}$. In other words, we removed the nodes corresponding to regions with a size below a given threshold value. The second has been performed by considering a non-increasing criterion, related to intervals of values for the same attribute $\mathcal{A}$, with a direct reconstruction policy. In other words, we preserved the nodes corresponding to regions with a size lying between two extremal values.

Some results of these two antiextensive filterings are depicted in Fig. 5. As stated above, the large size of remote sensing images, and their high resolution potentially result in classification maps that can be impaired by semantic noise. In this context, the simultaneous use of (i) an area criterion, and (ii) a set of hierarchically structured labels, allows us to efficiently carry out a mixed spatial / semantic denoising of the classification maps. In other words, it becomes possible to handle the trade-off between spatial and semantic filtering, e.g., by obtaining regions of homogeneous size and / or levels of semantic accuracy. Fig. 5(d) depicts a case where large semantic structures (i.e., meadow, agricultural, urban and water areas) are preserved. At the opposite, Fig. 5(e) illustrates a filtering result where smaller semantic structures (in this application the mineral, specific and housing areas) are preserved.

In order to evaluate the computational and spatial complexities related to the construction of the multivalued component-trees $\mathfrak{G}$, $\dot{\mathfrak{G}}$ and $\ddot{\mathfrak{G}}$ of an image $I^c : \Omega \to V$, we studied both the computation time required for building them and the number of nodes composing these structures by varying the size of the classification map. Experiments have been run on

an Intel® Core™ i7-3740QM running at 2.7 GHz with 8 GB of RAM. Fig. 6 presents the results of this complexity study. These graphs confirm that the computational and spatial complexities for constructing the multivalued component-trees are both quasi-linear with respect to the size of the images.

XI. CONCLUSION

In this article, it has been shown that when the image value space is organised in a tree fashion, the induced component-graphs, namely multivalued component-trees, also present a tree structure. In that case, it is possible to rely on the standard component-tree construction algorithms to efficiently build these multivalued component-trees. Moreover, the classical antiextensive filtering scheme initially proposed for component-trees [4], [8] can also be adapted, by developing an analogy between multivalued component-trees and standard "grey-level" component-trees.

Beyond the application example described in Sec. X that is devoted to process semantic images, it is important to notice that numerous relevant applications can also be considered for the processing of natural images, where the set V corresponds to physical value spaces, such as colour ones. As an example, it is possible to define a tree structure, e.g., on a RGB or HSV colour space V, by defining a cover $(V, \prec)$ as a spanning tree of the Hasse diagram induced by the canonical order $\leqslant$ of V. More precisely, this can be done by defining a valuation on the edges of this Hasse diagram. Such valuation can, in particular, carry a priori information associated to the applicative purposes. Based on such strategies, the multivalued component-trees can allow us to develop original extensions of the vectorial processing of colour images (see Sec. II-B).

In this rich context, the multivalued component-trees could also be considered to perform not only filtering, but also segmentation. More precisely, it could be relevant to consider standard optimal-cut segmentation paradigms, such as introduced in [43], and further considered in the framework of mathematical morphology (see [44] for a recent survey).

In the next part of this work, we will deal with the case where the component-graphs no longer present a tree structure, thus rising different algorithmic difficulties and potential non-determinism in image antiextensive filtering.

ACKNOWLEDGEMENTS

REFERENCES

[1] P. Salembier and J. Serra, "Flat zones filtering, connected operators, and filters by reconstruction," *IEEE T Image Process*, vol. 4, pp. 1153–1160, 1995.

[2] P. Salembier and M. H. F. Wilkinson, "Connected operators: A review of region-based morphological image processing techniques," *IEEE Signal Proc Mag*, vol. 26, pp. 136–157, 2009.

[3] L. Najman and H. Talbot, Eds., *Mathematical Morphology: From Theory to Applications*. ISTE/J. Wiley & Sons, 2010.

[4] P. Salembier, A. Oliveras, and L. Garrido, "Anti-extensive connected operators for image and sequence processing," *IEEE T Image Process*, vol. 7, pp. 555–570, 1998.

[5] B. Naegel and N. Passat, "Component-trees and multivalued images: A comparative study," in *ISMM*, ser. Lect Notes Comput Sc, vol. 5720, 2009, pp. 261–171.

[6] N. Passat and B. Naegel, "An extension of component-trees to partial orders," in *ICIP*, 2009, pp. 3981–3984.

[7] ——, "Component-trees and multivalued images: Structural properties," *J Math Imaging Vis*, vol. 49, pp. 37–50, 2014.

[8] R. Jones, "Connected filtering and segmentation using component trees," *Comput Vis Image Und*, vol. 75, pp. 215–228, 1999.

[9] C. Kurtz, B. Naegel, and N. Passat, "Multivalued component-tree filtering," in *ICPR (In Press)*, 2014.

[10] D. Wishart, "Mode analysis: A generalization of the nearest neighbor," in *Numerical Taxonomy*. Academic Press, 1969, pp. 282–319.

[11] J. A. Hartigan, "Statistical theory in clustering," *J Classif*, vol. 2, pp. 63–76, 1985.

[12] P. Hanusse and P. Guillataud, "Sémantique des images par analyse dendronique," in *RFIA*, 1991, pp. 577–588.

[13] L. Chen, M. W. Berry, and W. W. Hargrove, "Using dendronal signatures for feature extraction and retrieval," *Int J Imag Syst Tech*, vol. 11, pp. 243–253, 2000.

[14] J. Mattes and J. Demongeot, "Efficient algorithms to implement the confinement tree," in *DGCI*, ser. Lect Notes Comput Sc, vol. 1953, 2000, pp. 392–405.

[15] L. Najman and M. Couprie, "Building the component tree in quasi-linear time," *IEEE T Image Process*, vol. 15, pp. 3531–3539, 2006.

[16] C. Berger, T. Géraud, R. Levillain, N. Widynski, A. Baillard, and E. Bertin, "Effective component tree computation with application to pattern recognition in astronomical imaging," in *ICIP*, 2007, pp. 41–44.

[17] M. H. F. Wilkinson, H. Gao, W. H. Hesselink, J.-E. Jonker, and A. Meijster, "Concurrent computation of attribute filters on shared memory parallel machines," *IEEE T Pattern Anal*, vol. 30, pp. 1800–1813, 2008.

[18] E. Carlinet and T. Géraud, "A comparison of many max-tree computation algorithms," in *ISMM*, ser. Lect Notes Comput Sc, vol. 7883, 2013, pp. 73–84.

[19] G. K. Ouzounis and M. H. F. Wilkinson, "Mask-based second-generation connectivity and attribute filters," *IEEE T Pattern Anal*, vol. 29, pp. 990–1004, 2007.

[20] N. Passat, B. Naegel, F. Rousseau, M. Koob, and J.-L. Dietemann, "Interactive segmentation based on component-trees," *Pattern Recogn*, vol. 44, pp. 2539–2554, 2011.

[21] A. Dufour, O. Tankyevych, B. Naegel, H. Talbot, C. Ronse, J. Baruthio, P. Dokládal, and N. Passat, "Filtering and segmentation of 3D angiographic data: Advances based on mathematical morphology," *Med Image Anal*, vol. 17, pp. 147–164, 2013.

[22] J. Mattes, M. Richard, and J. Demongeot, "Tree representation for image matching and object recognition," in *DGCI*, ser. Lect Notes Comput Sc, vol. 1568, 1999, pp. 392–405.

[23] V. Mosorov, "A main stem concept for image matching," *Pattern Recogn Lett*, vol. 26, pp. 1105–1117, 2005.

[24] N. Alajlan, M. S. Kamel, and G. H. Freeman, "Geometry-based image retrieval in binary image databases," *IEEE T Pattern Anal*, vol. 30, pp. 1003–1013, 2008.

[25] E. R. Urbach, J. B. T. M. Roerdink, and M. H. F. Wilkinson, "Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images," *IEEE T Pattern Anal*, vol. 29, pp. 272–285, 2007.

[26] M. A. Westenberg, J. B. T. M. Roerdink, and M. H. F. Wilkinson, "Volumetric attribute filtering and interactive visualization using the max-tree representation," *IEEE T Image Process*, vol. 16, pp. 2943–2952, 2007.

[27] B. Naegel and L. Wendling, "A document binarization method based on connected operators," *Pattern Recogn Lett*, vol. 31, pp. 1251–1259, 2010.

[28] E. J. Breen and R. Jones, "Attribute openings, thinnings, and granulometries," *Comput Vis Image Und*, vol. 64, pp. 377–389, 1996.

[29] E. R. Urbach, N. J. Boersma, and M. H. F. Wilkinson, "Vector attribute filters," in *ISMM*, ser. Comput Imaging Vis, vol. 30, 2005, pp. 95–104.

[30] H. J. A. M. Heijmans, "Theoretical aspects of gray-level morphology," *IEEE T Pattern Anal*, vol. 13, pp. 568–582, 1991.

[31] E. Aptoula and S. Lefèvre, "A comparative study on multivariate mathematical morphology," *Pattern Recogn*, vol. 40, pp. 2914–2929, 2007.

[32] C. Ronse and V. Agnus, "Morphology on label images: Flat-type operators and connections," *J Math Imaging Vis*, vol. 22, pp. 283–307, 2005.

[33] V. Barnett, "The ordering of multivariate data," *J Roy Stat Soc A Sta*, vol. 139, pp. 318–354, 1976.

[34] J. Goutsias, H. J. A. M. Heijmans, and K. Sivakumar, "Morphological operators for image sequences," *Comput Vis Image Und*, vol. 62, pp. 326–346, 1995.

[35] H. Talbot, C. Evans, and R. Jones, "Complete ordering and multivariate mathematical morphology," in *ISMM*, 1998, pp. 27–34.

[36] E. Aptoula and S. Lefèvre, "On lexicographical ordering in multivariate mathematical morphology," *Pattern Recogn Lett*, vol. 29, pp. 109–118, 2008.

[37] J. Angulo, "Geometric algebra colour image representations and derived total orderings for morphological operators–Part I: Colour quaternions," *J Vis Commun Image R*, vol. 21, pp. 33–48, 2010.

[38] S. Velasco-Forero and J. Angulo, "Supervised ordering in $\mathbb{R}^p$: Application to morphological processing of hyperspectral images," *IEEE T Image Process*, vol. 20, pp. 3301–3308, 2011.

[39] L. Mazo, N. Passat, M. Couprie, and C. Ronse, "Digital imaging: A unified topological framework," *J Math Imaging Vis*, vol. 44, pp. 19–37, 2012.

[40] A. D. Gordon, "A review of hierarchical classification," *J Roy Stat Soc A Sta*, vol. 150, pp. 119–137, 1987.

[41] C. Kurtz, N. Passat, P. Gançarski, and A. Puissant, "Extraction of complex patterns from multiresolution remote sensing images: A hierarchical top-down methodology," *Pattern Recogn*, vol. 45, pp. 685–706, 2012.

[42] F. Petitjean, C. Kurtz, N. Passat, and P. Gançarski, "Spatio-temporal reasoning for the classification of satellite image time series," *Pattern Recogn Lett*, vol. 33, pp. 1805–1815, 2012.

[43] L. Guigues, J.-P. Cocquerez, and H. Le Men, "Scale-sets image analysis," *Int J Comput Vision*, vol. 68, pp. 289–317, 2006.

[44] J. Serra, "Tutorial on connective morphology," *IEEE J Sel Top Signal*, vol. 6, pp. 739–752, 2012.

Camille Kurtz obtained the MSc and PhD from Université de Strasbourg, France, in 2009 and 2012. He was a post-doctoral fellow at Stanford University, CA, USA, between 2012 and 2013. He is now an assistant professor at Université Paris Descartes, France. His scientific interests include image analysis, data mining, medical imaging and remote sensing.

Benoît Naegel obtained the MSc and PhD from Université Strasbourg 1, France, in 2000 and 2004. He was a research assistant at University of Applied Sciences Western Switzerland, Geneva, between 2005 and 2007, and an assistant professor at Université de Nancy, France, between 2007 and 2011. He is now an assistant professor at Université de Strasbourg. His scientific interests include mathematical morphology, document processing and medical imaging.

Nicolas Passat obtained the MSc and PhD from Université Strasbourg 1, France, in 2002 and 2005, and Habilitation from Université de Strasbourg, in 2011. He was a post-doctoral fellow at ESIEE-Paris, France, in 2006, and an assistant professor at Université de Strasbourg, between 2006 and 2012. He is now a full professor at Université de Reims Champagne-Ardenne, France. His scientific interests include mathematical morphology, discrete topology, medical imaging and remote sensing.