

Multivalued component-tree filtering

Camille Kurtz, Benoît Naegel, Nicolas Passat

▶ **To cite this version:**

Camille Kurtz, Benoît Naegel, Nicolas Passat. Multivalued component-tree filtering. International Conference on Pattern Recognition (ICPR), 2014, Stockholm, Sweden. pp.1008-1013, 10.1109/ICPR.2014.183 . hal-01695070

HAL Id: hal-01695070
https://hal.univ-reims.fr/hal-01695070

Submitted on 15 Feb 2018

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

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

Multivalued component-tree filtering

Camille Kurtz
Université Paris Descartes
LIPADE
Paris, France

Benoît Naegel
Université de Strasbourg
CNRS, ICube
Strasbourg, France

Nicolas Passat
Université de Reims Champagne-Ardenne
CReSTIC
Reims, France

***Abstract*—We introduce the new notion of multivalued component-tree, that extends the classical component-tree initially devoted to grey-level images, in the mathematical morphology framework. We prove that multivalued component-trees can model images whose values are hierarchically organized. We also show that they can be efficiently built from standard component-tree construction algorithms, and involved in antiextensive filtering procedures. The relevance and usefulness of multivalued component-trees is illustrated by an applicative example on hierarchically classified remote sensing images.**

I. Introduction

Connected operators [1], [2] gather image processing tools defined in the framework of mathematical morphology and involved in a wide spectrum of applications [3, Ch. 7]. In this context, the notion of component-tree [4] has received a specific attention. The component-tree is a hierarchical data structure that models a grey-level image by considering its binary level sets obtained from successive thresholdings. It is well-suited for grey-level image filtering and segmentation methods [5], [6], based on hypotheses related to the connectedness and extremal intensity of structures of interest, and relying on attribute-based [7], [8] or optimal cut strategies [9].

The success of the component-tree in the field of grey-level image processing, together with the increasing need for applications involving multivalued images [10], has motivated its extension to the case of such images, which can take their values in any – totally or partially – ordered sets [11]. To deal with this issue, a new notion of component-graph was recently introduced [12], [13]. By contrast with the component-tree, the component-graph is not a tree, in general. This structural property induces several algorithmic open issues.

We introduce intermediate data structures, called the *multivalued component-trees*. They constitute a subfamily of component-graphs, that still present a tree structure, and then extend the component-trees. They gather, in particular, the structural / algorithmic advantages of hierarchical data structures, but can be used beyond the domain of grey-level images.

This article is organised in three – theoretical, algorithmic and applicative – parts. We first characterise the value spaces that allow us to define multivalued component-trees (Sec. II). Then, we show how to efficiently build such trees, and involve them in an antiextensive filtering scheme similar to the one proposed in [4], [8] for grey-level images (Sec. III). We finally present an application of multivalued component-tree filtering on hierarchically classified remote sensing images (Sec. IV).

The research leading to these results has received funding from the French *Agence Nationale de la Recherche* (Grant Agreement ANR-12-MONU-0001).

II. Multivalued component-trees

Let Ω be a nonempty finite set. To define a notion of connectivity [14] on Ω, we consider the classical and versatile graph-based paradigm, generally used in discrete topology and digital image processing. More precisely, given an adjacency relation on Ω, we define for any $X \subseteq \Omega$, the (equivalence) connectedness relation as the reflexive-transitive closure of this adjacency on X. The set of all the connected components (*i.e.*, equivalence classes) of X, is noted $C[X]$.

Let V be a nonempty finite set equipped with an order relation $\leqslant$. We note $<$ and $\prec$ the strict and cover relations associated to $\leqslant$, respectively. We assume that $(V, \leqslant)$ admits a minimum, noted $\bot$.

Let us consider an image $I : \Omega \to V$. Each binary level set $\lambda_v(I) = \{x \in \Omega \mid v \leqslant I(x)\} \subseteq \Omega$ is divided into connected components, gathered into the partition $C[\lambda_v(I)]$. The union

$$\Phi = \bigcup_{v \in V} C[\lambda_v(I)] \tag{1}$$

of all these partitions can be equipped with the inclusion $\subseteq$.

Let us now suppose that $(V, \leqslant)$ is totally ordered, *i.e.*, I is a grey-level image. We can define the component-tree of I.

Definition 1 ([4]): The *component-tree* $\mathfrak{T}$ of I is the Hasse diagram of the partially ordered set $(\Phi, \subseteq)$.

Then, $\mathfrak{T}$ has a (rooted) tree structure, whose "root" is Ω, and "leaves" are the flat zones of I of locally maximal values. The cover associated to $\subseteq$ then provides a parent / children relation between the "nodes" of the tree. Before going on, let us clarify the actual meaning of "tree structure".

Definition 2: Let $(A, \sqsubseteq)$ be an ordered set. For any $a \in A$, we note $a^{\uparrow} = \{b \in A \mid a \sqsubseteq b\}$ and $a^{\downarrow} = \{b \in A \mid b \sqsubseteq a\}$. 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.

If we relax the totality constraint on $\leqslant$, then $\mathfrak{T}$ does no longer have a tree structure, in most cases. This has motivated the introduction of a more general notion of component-graph.

To define this notion, we enrich the set Φ, by assigning to each connected component the value of its level set. The obtained set of valued connected components is then defined as

$$\Theta = \bigcup_{v \in V} C[\lambda_v(I)] \times \{v\} \tag{2}$$

We then extend the inclusion relation on Φ by considering these values. We obtain the order relation $\trianglelefteq$ on Θ defined as

$$(X_1, v_1) \trianglelefteq (X_2, v_2) \Leftrightarrow (X_1 \subset X_2) \vee ((X_1 = X_2) \wedge (v_2 \leqslant v_1)) \quad (3)$$

We can finally define the notion of component-graph as follows.

Definition 3 ([13]): The *component-graph* $\mathfrak{G}$ of I is the Hasse diagram $(\Theta, \blacktriangleleft)$ of the partially ordered set $(\Theta, \trianglelefteq)$.

The component-graph can be declined into three variants, noted $\mathfrak{G}$, $\dot{\mathfrak{G}}$ and $\ddot{\mathfrak{G}}$ – of decreasing space complexity and richness – by also considering the following two subsets of Θ, namely

$$\dot{\Theta} = \{(X, v) \in \Theta \mid \forall (X, v') \in \Theta, v \not< v'\} \quad (4)$$

$$\ddot{\Theta} = \{(X, v) \in \Theta \mid \exists x \in X, v = I(x)\} \quad (5)$$

By contrast with the component-tree, the component-graph does not have a tree structure, in general. In this article, we focus on an intermediate notion, namely a subfamily of component-graphs:

- (*i*) more general than the component-trees, *i.e.*, defined for images that are not necessarily grey-level ones; but
- (*ii*) that present however a tree structure, and thus still have good properties for filtering purpose.

We call them *multivalued component-trees*.

The first question that arises is then: *When can we licitly define a multivalued component-tree on an image $I : \Omega \to V$?* The answer is provided by the following proposition.

Proposition 4: The component-graph $\mathfrak{G}$ of any image $I : \Omega \to V$ has a tree structure if and only if $(V, \leqslant)$ is a LPTOS.

Proof Let us first assume that $(V, \leqslant)$ is a LPTOS. Let $I : \Omega \to V$ be an image and $\mathfrak{G}$ its component-graph. Let $x \in \Omega$. Let $K = (X, v) \in \Theta$ such that $x \in X$ and $I(x) = v$. Let $K_1 = (X_1, v_1), K_2 = (X_2, v_2) \in K^{\uparrow}$. From Eq. (3), it comes $v_1, v_2 \leqslant v$. As $(V, \leqslant)$ is a LPTOS, we can assume $v_1 \leqslant v_2$. Then, Eq. (2) implies $X_2 \subseteq X_1$, and thus $K_2 \trianglelefteq K_1$. Finally, $(K^{\uparrow}, \trianglelefteq)$ is totally ordered, and $(\Theta, \trianglelefteq)$ is a UPTOS. Now, let us assume that for any $I : \Omega \to V$, the component-graph $\mathfrak{G}$ of I has a tree structure. As $(\Theta, \trianglelefteq)$ admits a maximum, namely $(\Omega, \bot)$, it is a UPTOS. Let $v \in V$ be a maximal element of $(V, \leqslant)$. Let $I : \Omega \to V$ 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. (2), it is isomorphic to $(v^{\downarrow}, \leqslant)$, that is then also totally ordered. Thus, $(V, \leqslant)$ is a LPTOS. ■

Multivalued component-trees can then be defined whenever the Hasse diagram of $(V, \leqslant)$ has a tree structure. In particular, this is true when $(V, \leqslant)$ is totally ordered. The multivalued component-tree is then a relevant extension of the component-tree, as stated by the following result.

Proposition 5 (From [13]): If $(V, \leqslant)$ is a totally ordered set, then two of the three variants of multivalued component-trees, namely $\dot{\mathfrak{G}}$ and $\ddot{\mathfrak{G}}$, are isomorphic to the component-tree.

From now on, we will assume that $(V, \leqslant)$ is a LPTOS.

III. Antiextensive filtering scheme

Based on this compatibility, multivalued component-trees inherit the (de)composition formula classically associated to component-trees, namely

$$I = \bigvee_{K \in \Theta}^{\leq} C_K \quad (6)$$

where $\bigvee$ is the supremum / maximum operator, $\leq$ is the pointwise order on functions induced by $\leqslant$, and for any $(X, v) \in \Theta$, $C_{(X,v)} : \Omega \to V$ is the cylinder function defined by $C_{(X,v)}(x) = v$ if $x \in X$ and $\bot$ otherwise.

In the framework of component-trees, this formula led to an antiextensive filtering scheme for grey-level images [4], [8]. This scheme, that we propose to extend to multivalued component-trees, consists of the following three successive steps (see also Diag. (7)):

- (*i*) construction of the multivalued component-tree $\mathfrak{G}$ associated to I;
- (*ii*) reduction of $\mathfrak{G}$, leading to a reduced multivalued component-tree $\widehat{\mathfrak{G}}$; and
- (*iii*) reconstruction of a filtered image $\widehat{I} \leq I$ induced by $\widehat{\mathfrak{G}}$.

$$\begin{array}{ccc} I & \longrightarrow & \widehat{I} \leq I \\ {\scriptstyle (i)}\downarrow & & \uparrow{\scriptstyle (iii)} \\ \mathfrak{G} & \xrightarrow{(ii)} & \widehat{\mathfrak{G}} \end{array} \quad (7)$$

One may notice that Step (*iii*) is indeed straightforward, by simply substituting the set of valued connected components $\widehat{\Theta}$ of $\widehat{\mathfrak{G}}$ to the set Θ in Eq. (6). We now explain how to efficiently handle the first two steps.

A. Multivalued component-tree construction

Many efforts have been successfully devoted to efficiently build component-trees [4], [15], [16], [17], [18], [19]. 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 a multivalued component-tree?* To answer this question, let us first consider the case of the multivalued component-tree $\dot{\mathfrak{G}} = (\dot{\Theta}, \dot{\blacktriangleleft})$ associated to $\dot{\Theta}$ (see Alg. 1).

We first define an extended set $\Omega_\Gamma \supseteq \Omega$, by adding a new point $\varepsilon_{\{x,y\}}$ for any adjacent points $x, y \in \Omega$ whose values are not comparable (lines 1–4). In the same time, we define an adjacency relation on Ω_Γ that is equal to the adjacency on Ω for any adjacent points $x, y \in \Omega$ whose values are comparable, and that replaces their adjacency by two adjacencies between x, $\varepsilon_{\{x,y\}}$ and $\varepsilon_{\{x,y\}}$, y, otherwise (lines 1–3,5).

We then extend I as a new image $I_\Gamma : \Omega_\Gamma \to V$ such that $I_\Gamma = I$ on Ω, while $I_\Gamma(\varepsilon_{\{x,y\}}) = \bigwedge^{\leqslant}\{I(x), I(y)\}$, namely the minimum of $I(x)$ and $I(y)$, for any $\varepsilon_{\{x,y\}} \in \Omega_\Gamma \setminus \Omega$.

From a topological point of view, the images I and I_Γ are deeply related. Indeed, we can define a bijection $\Gamma : \Theta \to \Theta_\Gamma$ such that $\Gamma((X, v)) = (X_\Gamma, v) \in \Theta_\Gamma$ with $X \subseteq X_\Gamma$. Its inverse function is, in particular, defined by $\Gamma^{-1}((X_\Gamma, v)) = (X_\Gamma \cap \Omega, v)$.

Algorithm 1: Construction of $\dot{\mathfrak{G}}$

Input: $I : \Omega \to V$ (multivalued image)
Input: $\frown \subseteq \Omega \times \Omega$ (adjacency inducing the connectedness on Ω)
Output: $\dot{\mathfrak{G}} = (\dot{\Theta}, \dot{\blacktriangleleft})$ (Hasse diagram of $(\dot{\Theta}, \trianglelefteq)$)

1 $(\Omega_\Gamma, \frown_\Gamma) \leftarrow (\Omega, \frown)$
2 **foreach** $x \frown y$ **do**
3 **if** $I(x) \not\leqslant I(y)$ **and** $I(y) \not\leqslant I(x)$ **then**
4 $\Omega_\Gamma \leftarrow \Omega_\Gamma \cup \{\varepsilon_{\{x,y\}}\}$
5 $\frown_\Gamma \leftarrow \frown_\Gamma \setminus \{(x, y)\} \cup \{(x, \varepsilon_{\{x,y\}}), (\varepsilon_{\{x,y\}}, y)\}$
6 **foreach** $x \in \Omega$ **do**
7 $I_\Psi(x) \leftarrow \psi(I(x))$
8 **foreach** $\varepsilon_{\{x,y\}} \in \Omega_\Gamma \setminus \Omega$ **do**
9 $I_\Psi(\varepsilon_{\{x,y\}}) \leftarrow \psi(\bigwedge^{\leqslant}\{I(x), I(y)\})$
10 **compute** $\dot{\mathfrak{G}}_\Psi = (\dot{\Theta}_\Psi, \dot{\blacktriangleleft}_\Psi)$ **for** $I_\Psi : \Omega_\Gamma \to \mathbb{N}$ **(see [19])**
11 $(\dot{\Theta}, \dot{\blacktriangleleft}) \leftarrow (\emptyset, \emptyset)$
12 **foreach** $K_\Psi \in \dot{\Theta}_\Gamma$ **do**
13 $\dot{\Theta} \leftarrow \dot{\Theta} \cup \{\Gamma^{-1}(\Psi^{-1}(K_\Psi))\}$
14 **foreach** $K_\Psi \mathrel{\dot{\blacktriangleleft}_\Psi} K'_\Psi$ **do**
15 $\dot{\blacktriangleleft} \leftarrow \dot{\blacktriangleleft} \cup \{(\Gamma^{-1}(\Psi^{-1}(K_\Psi)), \Gamma^{-1}(\Psi^{-1}(K'_\Psi)))\}$

Algorithm 2: Construction of $\mathfrak{G}$ from $\dot{\mathfrak{G}}$

Input: $\dot{\mathfrak{G}} = (\dot{\Theta}, \dot{\blacktriangleleft})$ (Hasse diagram of $(\dot{\Theta}, \trianglelefteq)$)
Output: $\mathfrak{G} = (\Theta, \blacktriangleleft)$ (Hasse diagram of $(\Theta, \trianglelefteq)$)

1 $(\Theta, \blacktriangleleft) \leftarrow (\dot{\Theta}, \emptyset)$
2 **foreach** $(X, v) \mathrel{\dot{\blacktriangleleft}} (X', v')$ **do**
3 **if** $v' \prec v$ **then**
4 $\blacktriangleleft \leftarrow \blacktriangleleft \cup \{((X, v), (X', v'))\}$
5 **else**
6 **foreach** $v' \prec w' \prec v$ **do**
7 $\Theta \leftarrow \Theta \cup \{(X, w')\}$
8 **for** $v' \prec w \prec v$ **do**
9 $\blacktriangleleft \leftarrow \blacktriangleleft \cup \{((X, w), (X', v'))\}$
10 **foreach** $w \leqslant w' \prec w'' \leqslant v$ **do**
11 $\blacktriangleleft \leftarrow \blacktriangleleft \cup \{((X, w''), (X, w'))\}$

Algorithm 3: Construction of $\ddot{\mathfrak{G}}$ from $\dot{\mathfrak{G}}$

Input: $I : \Omega \to V$ (multivalued image)
Input: $\dot{\mathfrak{G}} = (\dot{\Theta}, \dot{\blacktriangleleft})$ (Hasse diagram of $(\dot{\Theta}, \trianglelefteq)$)
Output: $\ddot{\mathfrak{G}} = (\ddot{\Theta}, \ddot{\blacktriangleleft})$ (Hasse diagram of $(\ddot{\Theta}, \trianglelefteq)$)

1 $(\ddot{\Theta}, \ddot{\blacktriangleleft}) \leftarrow (\dot{\Theta}, \dot{\blacktriangleleft})$
2 **foreach** $K = (X, v) \in \ddot{\Theta}$ **do**
3 **if** $\forall x \in X, I(x) \neq v$ **then**
4 $\ddot{\Theta} \leftarrow \ddot{\Theta} \setminus \{K\}$
5 **for** $K' \mathrel{\ddot{\blacktriangleleft}} K$ **do**
6 $\ddot{\blacktriangleleft} \leftarrow \ddot{\blacktriangleleft} \setminus \{(K, K')\}$
7 **foreach** $K'' \mathrel{\ddot{\blacktriangleleft}} K$ **do**
8 $\ddot{\blacktriangleleft} \leftarrow \ddot{\blacktriangleleft} \setminus \{(K'', K)\} \cup \{(K'', K')\}$

Property 6: The bijection Γ induces an isomorphism between $(\dot{\Theta}, \trianglelefteq)$ and $(\dot{\Theta}_\Gamma, \trianglelefteq_\Gamma)$. In particular, $\dot{\mathfrak{G}}$ is isomorphic to $\dot{\mathfrak{G}}_\Gamma$.

Let $\psi : V \to \mathbb{N}$ be the function defined by $\psi(v) = |v^\downarrow|$, namely the number of elements of $v^\downarrow$. It induces a homomorphism from $(V, \leqslant)$ to the totally ordered set $(\mathbb{N}, \leqslant_\mathbb{N})$. If $(V, \leqslant)$ is not totally ordered, then ψ is not injective. However, this is not a problem when considering $I_\Gamma : \Omega_\Gamma \to V$, since any adjacent points have now comparable values.

Then, let us consider the grey-level image $I_\Psi : \Omega_\Gamma \to \mathbb{N}$, defined as $I_\Psi = \psi \circ I_\Gamma$ (lines 6–9). We can define the bijection $\Psi : \dot{\Theta}_\Gamma \to \dot{\Theta}_\Psi$ such that $\Psi((X_\Gamma, v)) = (X_\Gamma, \psi(v))$. Its inverse function is defined by $\Psi^{-1}((X_\Gamma, v_\Psi)) = (X_\Gamma, \bigwedge^{\leqslant}\{I(x) \mid x \in X_\Gamma\})$.

Property 7: The bijection Ψ induces an isomorphism between $(\dot{\Theta}_\Gamma, \trianglelefteq_\Gamma)$ and $(\dot{\Theta}_\Psi, \trianglelefteq_\Psi)$. In particular, $\dot{\mathfrak{G}}_\Gamma$ is isomorphic to $\dot{\mathfrak{G}}_\Psi$.

As $I_\Psi : \Omega_\Gamma \to \mathbb{N}$ is actually a grey-level image, $\dot{\mathfrak{G}}_\Psi$ is (isomorphic to) a component-tree (Prop. 5). Thus, it can be built from any standard component-tree construction algorithm (line 10). The multivalued component-tree $\dot{\mathfrak{G}}$ can then be straightforwardly retrieved from $\dot{\mathfrak{G}}_\Psi$ thanks to Ψ^{-1} and Γ^{-1} (lines 11–15).

The pre- and post-processings (lines 1–9 and 11–15) both present a computational cost of $O(|\Omega| + |\frown|)$. It is proved in [16] that a component-tree can be built in quasi-linear time with respect to the size of the image. Thus, the computational cost for building $\dot{\mathfrak{G}}$ is quasi-linear with respect to $|\Omega| + |\frown|$. One may notice that we generally have $|\frown| = O(|\Omega|)$; in that frequent case, the cost is quasi-linear with respect to $|\Omega|$.

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

Let us first consider the case of $\mathfrak{G}$. For any link $(X, v) \mathrel{\dot{\blacktriangleleft}} (X', v')$ such that there exists a value $w' \in V$ verifying $v' \prec w' \prec v$, we remove this link and we update $\dot{\mathfrak{G}}$ as follows: (*i*) we add to $\dot{\Theta}$ all the valued connected components (X, w') such that $v' \prec w' \prec v$; and (*ii*) we create the respective links between all these new successive valued connected components, with respect to $\prec$, plus the two links with (X, v) and (X', v'). (See [13, Prop. 12] for a formal justification of these modifications.) This process is formalised in Alg. 2.

Let us now consider the case of $\ddot{\mathfrak{G}}$. For any $K = (X, v) \in \dot{\Theta}$ such that for all $x \in X$, $I(x) \neq v$, we remove K from $\dot{\Theta}$ and we update $\dot{\mathfrak{G}}$ as follows: (*i*) we remove the unique edge $(K, K') \in \dot{\blacktriangleleft}$; and (*ii*) we replace any edge $(K'', K) \in \dot{\blacktriangleleft}$ by (K'', K'). This process is formalised in Alg. 3.

B. Multivalued component-tree reduction

The second step of the scheme consists of reducing $\mathfrak{G}$. This implies to choose a subset of nodes $\widehat{\Theta} \subseteq \Theta$. Similarly to the case of component-tree filtering [4], [8], this choice relies on:

- a selection criterion, *i.e.*, a Boolean predicate $\rho : \Theta \to \mathbb{B}$ that indicates if the nodes satisfy a required property; and
- a reduction policy which is combined with ρ to determine which parts of the component-graph should be preserved or discarded.

If ρ models an increasing criterion, *i.e.*, if $K \trianglelefteq K' \Rightarrow (\rho(K) \Rightarrow \rho(K'))$, then the way to reduce $\mathfrak{G}$ is straightforward (in particular, the following reduction policies are equivalent).

However, if ρ models a non-increasing criterion, 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 8 (From [4]): Let $(V, \leqslant)$ be a LPTOS. Let $\widehat{\Theta} \subseteq \Theta$. Let $\rho : \Theta \to \mathbb{B}$ be a selection criterion. The three reduction policies min, direct and max are characterized by the following axiomatic (recursive) definitions of the subsets $\widehat{\Theta}_{\text{min}}, \widehat{\Theta}_{\text{direct}}, \widehat{\Theta}_{\text{max}} \subseteq \widehat{\Theta}$.

$$(\Omega, \bot) \in \widehat{\Theta}_{\text{min}}, \widehat{\Theta}_{\text{direct}}, \widehat{\Theta}_{\text{max}} \tag{8}$$

$$\rho(K) \wedge (K' \blacktriangleright K \wedge K' \in \widehat{\Theta}_{\text{min}}) \Longrightarrow K \in \widehat{\Theta}_{\text{min}} \tag{9}$$

$$\rho(K) \Longrightarrow K \in \widehat{\Theta}_{\text{direct}} \tag{10}$$

$$\rho(K) \vee (K' \blacktriangleleft K \wedge K' \in \widehat{\Theta}_{\text{max}}) \Longrightarrow K \in \widehat{\Theta}_{\text{max}} \tag{11}$$

Property 9: We have the following inclusion relations between the nine reduced sets of nodes of Θ.

$$\begin{array}{ccccc}
\widehat{\Theta}_{\text{max}}(\ddot{\mathfrak{G}}) & \xrightarrow{\subseteq} & \widehat{\Theta}_{\text{max}}(\dot{\mathfrak{G}}) & \xrightarrow{\subseteq} & \widehat{\Theta}_{\text{max}}(\mathfrak{G}) \\
\subseteq\uparrow & & \subseteq\uparrow & & \subseteq\uparrow \\
\widehat{\Theta}_{\text{direct}}(\ddot{\mathfrak{G}}) & \xrightarrow{\subseteq} & \widehat{\Theta}_{\text{direct}}(\dot{\mathfrak{G}}) & \xrightarrow{\subseteq} & \widehat{\Theta}_{\text{direct}}(\mathfrak{G}) \\
\subseteq\uparrow & & \subseteq\uparrow & & \subseteq\uparrow \\
\widehat{\Theta}_{\text{min}}(\ddot{\mathfrak{G}}) & & \widehat{\Theta}_{\text{min}}(\dot{\mathfrak{G}}) & & \widehat{\Theta}_{\text{min}}(\mathfrak{G})
\end{array} \tag{12}$$

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

IV. Experiments and Results

A. Synthetic data

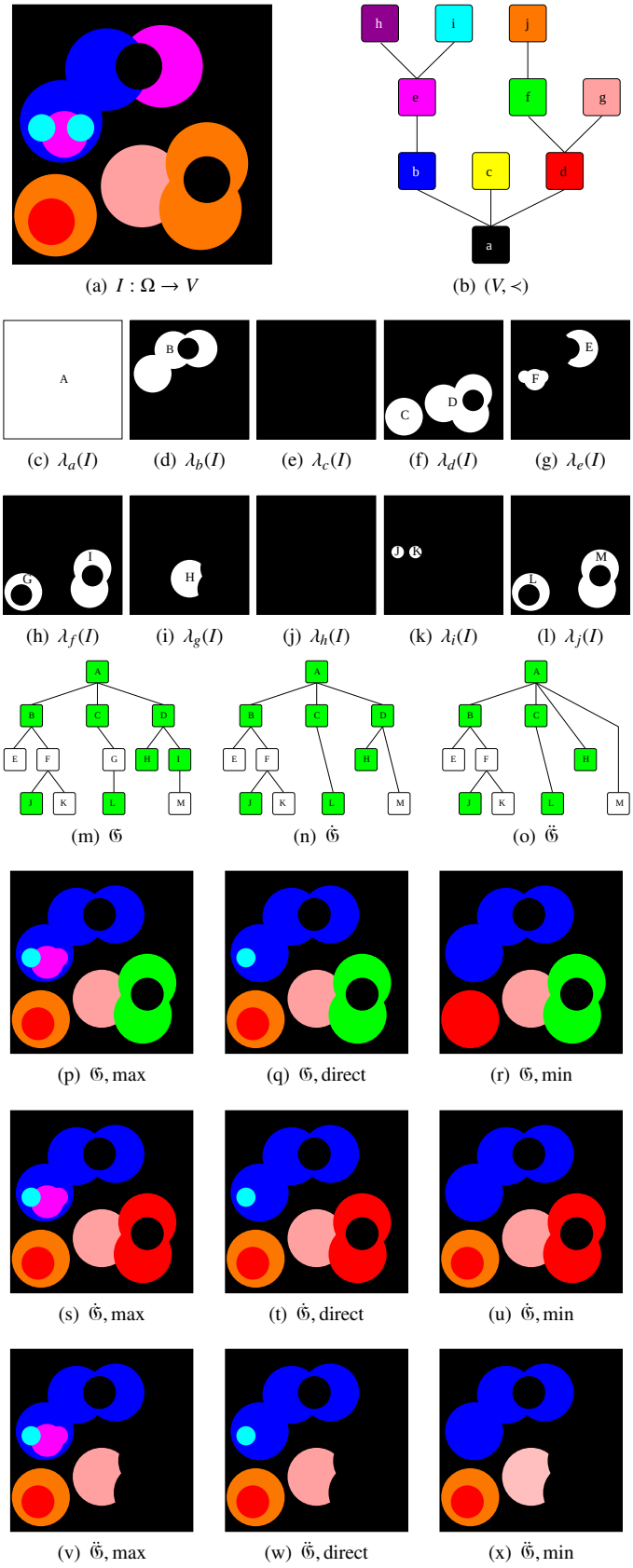
Before considering a real application, let us illustrate the effects of the choice of the multivalued component-tree and the reduction policy, when filtering a multivalued image.

To this end, we consider the synthetic example of Fig. 1, that provides a multivalued image $I : \Omega \to V$ (Fig. 1(a)) taking its values in a LPTOS $(V, \leqslant)$ (Fig. 1(b)). This image is associated to three distinct multivalued component-trees $\mathfrak{G} = (\Theta, \blacktriangleleft)$, $\dot{\mathfrak{G}} = (\dot{\Theta}, \dot{\blacktriangleleft})$, $\ddot{\mathfrak{G}} = (\ddot{\Theta}, \ddot{\blacktriangleleft})$ (Fig. 1(m–o)), with $\Theta = \{\text{A}, \text{B}, \ldots, \text{M}\}$ (Fig. 1(c–l)).

Let us now suppose that for a given selection criterion $\rho : \Theta \to \mathbb{B}$, we have $\rho^{-1}(\{\text{true}\}) = \{\text{A}, \text{B}, \text{C}, \text{D}, \text{H}, \text{I}, \text{J}, \text{L}\}$. By reducing the three multivalued component-trees, with the three policies max, direct, min, we obtain nine reduced multivalued component-trees from which we can reconstruct nine filtered images, illustrated in (Fig. 1(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. Hierarchically classified image filtering

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 [20]. Such hierarchical maps – that can be obtained from supervised or unsupervised classification – are widely considered in the context



(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) $\mathfrak{G}$, max (q) $\mathfrak{G}$, direct (r) $\mathfrak{G}$, min

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

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

Fig. 1. (a) An image $I : \Omega \to V$ with $V = \{a, \ldots, j\}$. (b) Hasse diagram of $(V, \leqslant)$. (c–l) Thresholded images $\lambda_v(I)$. (m–o) Multivalued component-trees of I. (p–x) Filtered images for $\rho(\star) = \text{true}$ for $\star \in \{\text{A}, \text{B}, \text{C}, \text{D}, \text{H}, \text{I}, \text{J}, \text{L}\}$ (see green nodes in (m–o)), depending on the chosen tree and reduction policy.

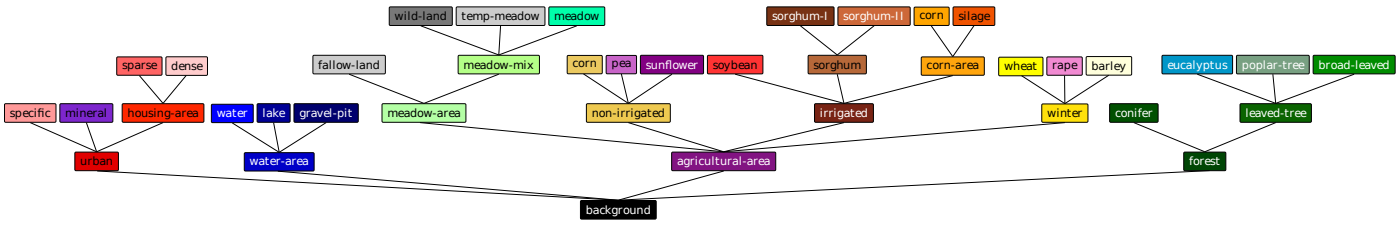
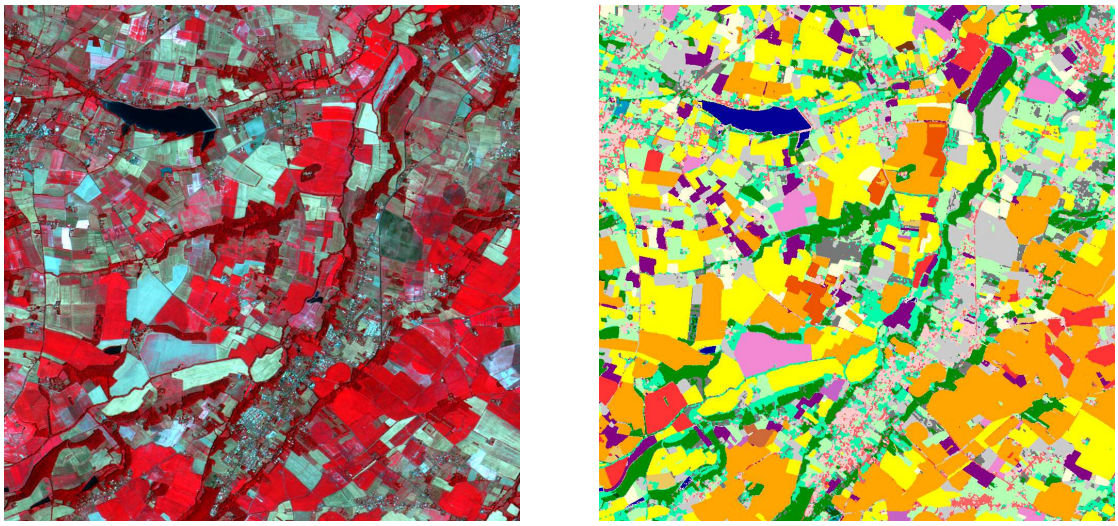
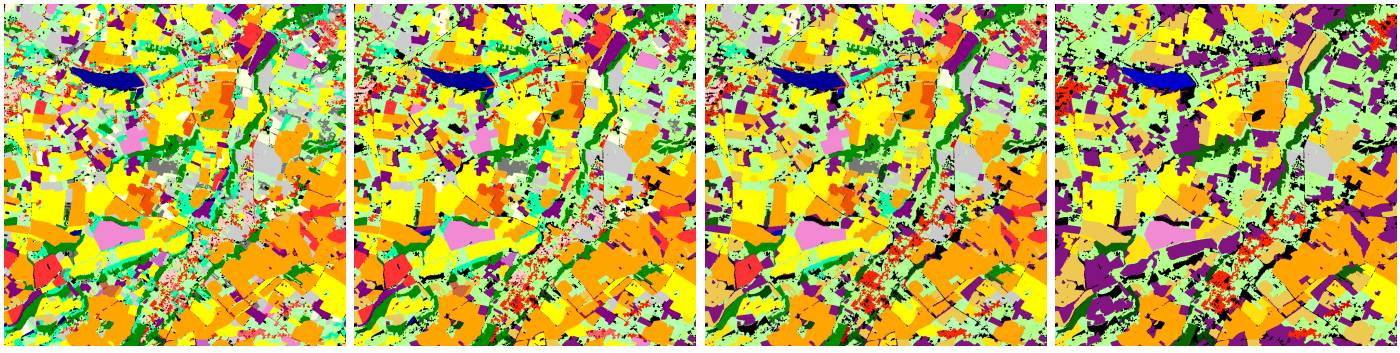


Fig. 2. Hasse diagram of a LPTOS $(V, \leqslant)$ where V is composed of 40 semantic labels, and $\leqslant$ provides the generalisation / specialisation links between them.

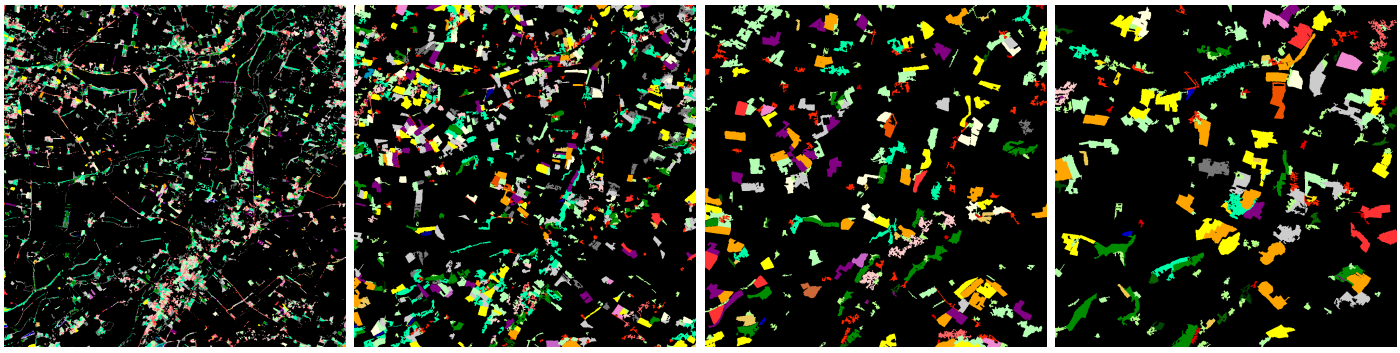


(a) Satellite image $S : \Omega \to \mathbb{R}^4$ (b) Classification map $I : \Omega \to V$

Fig. 3. (a) Satellite image (FORMOSAT-2), defined on a set Ω of 1000×1000 pixels. Spatial resolution: 8 m $\times$ 8 m. Spectral resolution: 4 bands (near infrared, red, green, blue). (b) Classification map $I : \Omega \to V$ obtained from (a) (courtesy J. Inglada, D. Ducrot, C. Marais-Sicre, O. Hagolle and M. Huc, CESBIO).



(a) $\tau_{min} = 10\,000$ m^2 (b) $\tau_{min} = 50\,000$ m^2 (c) $\tau_{min} = 100\,000$ m^2 (d) $\tau_{min} = 500\,000$ m^2



(e) $[\tau_{min}, \tau_{max}] = [0, 10\,000]$ m^2 (f) $[\tau_{min}, \tau_{max}] = [10\,000, 50\,000]$ m^2 (g) $[\tau_{min}, \tau_{max}] = [50\,000, 100\,000]$ m^2 (h) $[\tau_{min}, \tau_{max}] = [100\,000, 200000]$ m^2

Fig. 4. Antiextensive filtering of I (Fig. 3(b)), based on an area attribute $\mathcal{A}$. While I takes its values within the 26 leaves of $(V, <)$ (Fig. 2), the filtered images can also contain other values, then modeling areas with less specific semantics. The processing of I (construction and reduction of the multivalued component-tree) requires 1.04 s CPU on an Intel® Core™2 Quad running at 2.4 GHz with 8 GB of RAM. (a–d) Results for $\tau_{min} < \mathcal{A}$. (e–h) Results for $\mathcal{A} \in [\tau_{min}, \tau_{max}]$.

of remote sensing, for instance when analysing multiresolution [21] or multitemporal satellite images [22].

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" (*i.e.*, the minimal elements) are the most precise labels – that correspond to the smallest details of related images – while the "root" (*i.e.*, the maximum) is the most general label – that is valid within the whole images.

In our application, we define a set V of labels related to the semantic elements of agricultural areas, illustrated in Fig. 2. 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. 3(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, thus leading to the classification map $I : \Omega \to V$ depicted in Fig. 3(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 semantic images.

Based on the multivalued component-tree $\dot{\mathfrak{G}}$ of $I : \Omega \to V$, 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 valued connected components 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 valued connected components corresponding to regions with a size lying between two extremal values.

Some results of these two antiextensive filterings are depicted in Fig. 4. As stated above, the large size of satellite images, and their high resolution results 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 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. 4(d) depicts an example of this trade-off where only the largest semantic structures (*i.e.*, urban and agricultural areas, forest) are preserved. At the opposite, Fig. 4(e) illustrates a filtering result where only small semantic structures (in this application the dense housing areas) are preserved.

These results illustrate the usefulness of multivalued component-trees for filtering. Beyond the previous example, devoted to semantic images, other applications can also be considered for natural images. Indeed, it is possible to define a tree structure, *e.g.*, on a RGB or HSV space V, by defining a cover $(V, \prec)$ as a spanning tree of the Hasse diagram induced by the canonical order $\leqslant$ of V. The multivalued component-trees can then allow us to develop original extensions of the vectorial processing of colour images. In this context, segmentation by optimal cut [9] also constitutes a short term perspective.

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] 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.

[6] 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.

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

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

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

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

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

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

[13] N. Passat and B. Naegel, "Component-trees and multivalued images: Structural properties," *J Math Imaging Vis*, vol. 49, pp. 37–50, 2014.

[14] U. Braga-Neto and J. K. Goutsias, "A theoretical tour of connectivity in image processing and analysis," *J Math Imaging Vis*, vol. 19, pp. 5–31, 2003.

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

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

[17] 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.

[18] 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.

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

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

[21] 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.

[22] 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.