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

Component-graph construction

Nicolas Passat · Benoît Naegel · Camille Kurtz

the date of receipt and acceptance should be inserted later

Abstract Component-trees are classical tree structures for grey-level image modelling. Component-graphs are defined as a generalization of component-trees to images taking their values in any (totally or partially) ordered sets. Similarly to component-trees, component-graphs are a lossless image model; then, they can allow for the development of various image processing approaches. However, component-graphs are not trees, but directed acyclic graphs. This makes their construction non-trivial, leading to non-linear time cost and resulting in non-linear space data structures. In this theoretical article, we discuss the notion(s) of component-graph, and we propose a strategy for their efficient building and representation, which are necessary conditions for further involving them in image processing approaches.

Keywords Component-graph · algorithmics · mathematical morphology · multivalued images

1 Introduction

The component-graph is a hierarchical image model developed in the framework of mathematical morphology [1]. Most hierarchical models proposed in this context are trees, i.e. rooted, non-directed, connected, acyclic graphs. Well-known examples of such trees are, for instance, the component-tree [2], the tree of shapes [3] or the binary partition tree [4]. By contrast, the component-graph is not a tree, but a directed acyclic graph (it shares this property with other models, such as component hypertrees [5] and directed component hierarchies [6]). This data structure is then more complex to build, store and manipulate. However, it is also more powerful, since it allows us to generalize the paradigm of component-tree not only to images taking their values in totally ordered (i.e. grey-level) sets, but in any (possibly partially) ordered sets. It then encompasses the spectrum of multivalued imaging.

After a preliminary study of the relations between component-trees and multivalued images [7], the notion of component-graph was introduced in [8]. A structural discussion was proposed in [9]. From an algorithmic point of view, first strategies for building component-graphs were investigated. Except in a specific case — where the partially ordered set of values is structured itself as a tree [10]— these first attempts emphasized a high computational cost of the construction process, and a high spatial cost of the directed acyclic graph explicitly modelling a component-graph [11,12]. By relaxing certain constraints, leading to an improved complexity, the notion of component-graph was involved in the development of an efficient extension of tree of shapes to multivalued images [13]. From an applicative point of view, the notion of component-graph, coupled with the recent notion of shaping [14], also led to preliminary, yet promising, results for multimodal image processing [15,16].

The purpose of this theoretical study is to describe different variants of component-graphs, and to propose an algorithmic scheme for efficiently building the most relevant ones. By side effect, we also discuss the way to efficiently store the resulting component-graph. This

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

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

Camille Kurtz
Université Paris-Descartes, LIPADE, France
E-mail: camille.kurtz@parisdescartes.fr

article is an extended and improved version of the conference paper [17].

It is organised as follows. Sections 2 and 3 provide notations and basic notions. Sections 4 and 5 provide a discussion on various kinds of component-graphs. In particular, we motivate our choice of working in priority with a specific family called the *strong component-graphs*. Sections 6–11 constitute the main contribution of the article. They discuss image preprocessing, data-structure modelling and construction steps for the actual building and storage of a component-graph. Section 12 provides a complementary discussion about the space versus time cost trade-off to be handled for efficiently manipulating the component-graph.

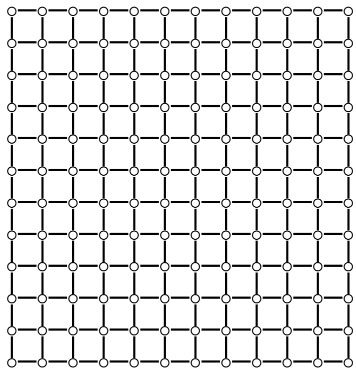

Fig. 1 A graph $(\Omega, \frown)$ where Ω is composed of 144 elements depicted by disks, whereas the adjacency relation $\frown$ between these elements is depicted by straight segments. This graph has the same structure as a 12×12 part of $\mathbb{Z}^2$ endowed with the standard 4-adjacency relation.

2 Notations

We use the same notations as in [9–12].

The inclusion (resp. strict inclusion) relation 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. If $\mathcal{P} \subseteq 2^X$ is a partition of X, we write $X = \bigsqcup \mathcal{P}$.

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 $F^{-1} : Y \to X$ its associated inverse function.

Let $\frown$ be a (binary) relation on a set X. The restriction of $\frown$ to a subset $Y \subseteq X$ will generally still be noted $\frown$.

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

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

For any symbol further used to denote an order relation ($\subseteq$, $\leq$, $\unlhd$, etc.), the inverse symbol ($\supseteq$, $\geq$, $\unrhd$, etc.) denotes the associated dual order, while the symbol without lower bar ($\subset$, $<$, $\lhd$, 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 $x^\uparrow = \{y \in X \mid y \geqslant x\}$ and $x^\downarrow = \{y \in X \mid y \leqslant x\}$, namely the sets of the elements greater and lower than x, respectively. If $Y \subseteq X$, the sets 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. If Y is defined as $\{x \mid p(x)\}$ where p is a Boolean predicate, we will sometimes note $\curlyvee^{\leqslant}_{p(x)} x$, instead of $\curlyvee^{\leqslant} Y$; the same remark holds for $\curlywedge^{\leqslant}$, $\bigvee^{\leqslant}$, $\bigwedge^{\leqslant}$, $\bigcup$, $\bigcap$, $\bigsqcup$.

The symbol $\leq$ will be used to denote two distinct orders: the canonical order on $\mathbb{Z}$ and the pointwise order on functions; the context of use allows the reader to unambiguously associate the correct semantics to each occurrence of the symbol.

3 Basic notions

3.1 Graph

Let Ω be a nonempty finite set. Let $\frown$ be an adjacency (i.e. irreflexive, symmetric) relation on Ω. The set $(\Omega, \frown)$ is then a non-directed graph. Let $X \subseteq \Omega$ be a subset of Ω. The reflexive-transitive closure of (the restriction of) $\frown$ on X induces the connectedness relation on X. It is an equivalence relation, and the set of the equivalence classes of X, called connected components, is noted $\mathcal{C}[X]$. Without loss of generality, we assume that $(\Omega, \frown)$ is connected, i.e. $\mathcal{C}[\Omega] = \{\Omega\}$.

Figure 1 illustrates a graph $(\Omega, \frown)$ that represents a 12×12 part of $\mathbb{Z}^2$ endowed with the standard 4-adjacency relation.

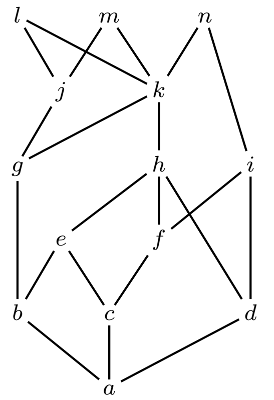

Fig. 2 Hasse diagram of a set $V = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n\}$, equipped with a partial order relation $\leqslant$. The straight segments correspond to the cover relation $\prec$ associated to $\leqslant$. The greater a value, the higher in the graph. For instance, we have $h \prec k$, since k is at a higher position than h. In particular, as $a = \bot$ is the minimum of $(V, \leqslant)$, it is at the lowest position. This ordered set does not have any maximum, but three maximal values l, m and n.

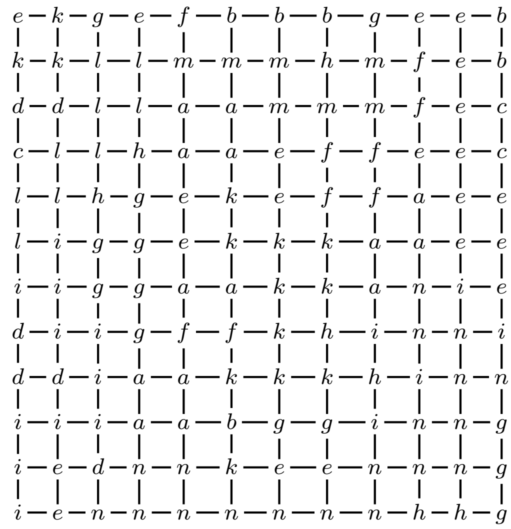

Fig. 3 An image $I : \Omega \to V$, where $(\Omega, \frown)$ is the graph of Figure 1 and $(V, \leqslant)$ is the ordered set of Figure 2. For the sake of readability, each element x of Ω is replaced by $I(x)$. This image will be used for all the further illustrations of the article.

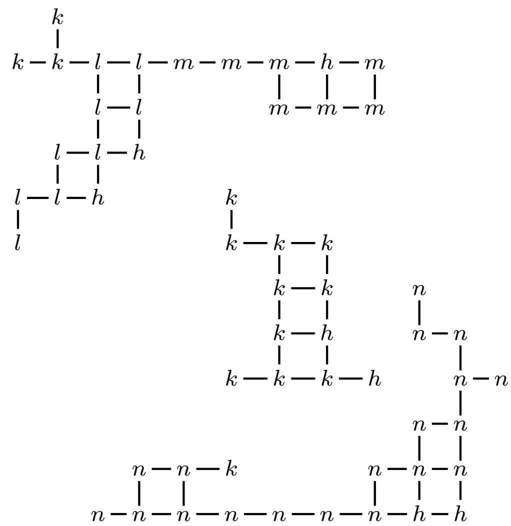

Fig. 4 The image I of Figure 3, thresholded at value h. More precisely, we show here the subgraph $(\lambda_h(I), \frown)$ (Equation (1)). The values $I(x)$ at the remaining points $x \in \lambda_h(I)$ are provided, for easing the comparison with Figure 3. We can observe that $(\lambda_h(I), \frown)$ has three connected components.

3.2 Image

Let Ω be a non-directed graph, such as defined above. 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 $I : \Omega \to V$. The sets Ω and V are called the support and the value space 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.

Figure 2 exemplifies a partially ordered set $(V, \leqslant)$ composed of 14 values. More precisely, it illustrates the Hasse diagram $(V, \prec)$ associated to $(V, \leqslant)$. Figure 3 illustrates an image I defined on the support Ω of Figure 1, and taking its values in the ordered set $(V, \leqslant)$ of Figure 2.

3.3 Level-set image (de)composition

Let $X \subseteq \Omega$ and $v \in V$. The thresholding function λ_v is defined by

$$\left| \begin{array}{l} \lambda_v : V^{\Omega} \to 2^{\Omega} \\ \quad\;\; I \;\;\mapsto \{x \in \Omega \mid I(x) \geqslant v\} \end{array} \right. \qquad (1)$$

An example of thresholded image is illustrated in Figure 4.

The cylinder function $C_{(X,v)}$ is defined by

$$\left| \begin{array}{l} C_{(X,v)} : \Omega \to V \\ \qquad\quad x \mapsto \begin{cases} v & \text{if } x \in X \\ \bot & \text{otherwise} \end{cases} \end{array} \right. \qquad (2)$$

An image $I : \Omega \to V$ can be decomposed into cylinder functions induced by thresholding operations and, symmetrically, I can be reconstructed by composition of these cylinder functions

$$I = \bigvee_{v \in V}^{\leq} \bigvee_{X \in \mathcal{C}[\lambda_v(I)]}^{\leq} C_{(X,v)} \qquad (3)$$

4 Valued connected components and orders

4.1 Valued connected components

We note Ψ the set of all the connected components obtained from all the thresholdings of I

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

By assigning to $X \in \Psi$ a value $v \in V$ such that X is a connected component of the level set $\lambda_v(I)$, we can define a notion of *valued connected component*.

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

$$\Theta = \bigcup_{v \in V} \mathcal{C}[\lambda_v(I)] \times \{v\} \tag{5}$$

4.2 Orders on valued connected components

In the case of grey-level images, the connected components of Ψ are equipped with the inclusion relation $\subseteq$. The component-tree is then defined as the Hasse diagram of the partially ordered set $(\Psi, \subseteq)$.

The purpose of component-graphs is to generalize this model to the case of images taking their values in sets which are not canonically equipped with total orders; this is for instance the case of multivalued images. This requires to define a relevant order relation $\trianglelefteq$ on the set Θ of valued connected components, and to ensure that this relation remains compliant with $\subseteq$ in the case where $\leqslant$ is a total order on V. Under such conditions, the component-graph, defined as the Hasse diagram of $(\Theta, \trianglelefteq)$, will be a relevant generalization of the component-tree.

Practically, there exist several ways to define a relation $\trianglelefteq$ on Θ that takes into account the support of the valued connected components and / or their value, i.e. that involves $\subseteq$ and / or $\leqslant$.

In particular, five variants of $\trianglelefteq$ can be proposed, in first intention, by composing $\subseteq$ and / or $\leqslant$ via standard policies (namely conjunction, lexicography, projection)

$$(X, v) \trianglelefteq_1 (Y, w) \Leftrightarrow X \subseteq Y \tag{6}$$

$$(X, v) \trianglelefteq_2 (Y, w) \Leftrightarrow (X \subset Y) \vee (X = Y \wedge w \leqslant v) \tag{7}$$

$$(X, v) \trianglelefteq_3 (Y, w) \Leftrightarrow X \subseteq Y \wedge w \leqslant v \tag{8}$$

$$(X, v) \trianglelefteq_4 (Y, w) \Leftrightarrow (w < v) \vee (w = v \wedge X \subseteq Y) \tag{9}$$

$$(X, v) \trianglelefteq_5 (Y, w) \Leftrightarrow w \leqslant v \tag{10}$$

In particular, for any $K, K' \in \Theta$, we have

$$K \trianglelefteq_3 K' \Rightarrow K \trianglelefteq_2 K' \Rightarrow K \trianglelefteq_1 K' \tag{11}$$

$$K \trianglelefteq_3 K' \Rightarrow K \trianglelefteq_4 K' \Rightarrow K \trianglelefteq_5 K' \tag{12}$$

However, on the one hand, neither $\trianglelefteq_1$ nor $\trianglelefteq_5$ are order relations. Indeed, they are reflexive and transitive, by not antisymmetric, in general. On the other hand, $\trianglelefteq_4$ is an order relation, but it does not relevantly take into account the support information carried by the elements of Θ. Indeed, two valued connected components are mainly compared with respect to their respective values. Their supports are only considered for a common value, and the condition $X \subseteq Y$, in Equation (9), simply rewrites as $X = Y$, thus ensuring antisymmetry.

Finally, only two relevant order relations remain, namely $\trianglelefteq_2$ and $\trianglelefteq_3$. In the sequel, they will be called *strong* and *weak orders* on Θ, respectively.

Definition 2 (Strong order, weak order) *The* strong order $\trianglelefteq_s$ *and the* weak order $\trianglelefteq_w$ *on Θ are defined as follows*

$$(X, v) \trianglelefteq_w (Y, w) \Leftrightarrow (X \subset Y) \vee (X = Y \wedge w \leqslant v) \tag{13}$$

$$(X, v) \trianglelefteq_s (Y, w) \Leftrightarrow X \subseteq Y \wedge w \leqslant v \tag{14}$$

The weak order is indeed a lexicographic order, whereas the strong order is a conjunctive order.

From Equation (12), we have the following property.

Property 1 *Let $K, K' \in \Theta$. We have*

$$K \trianglelefteq_s K' \Rightarrow K \trianglelefteq_w K' \tag{15}$$

In other words, the strong order relation implies the weak one. The reverse is not true, in general.

5 Component-graphs

In the sequel, the notation $\trianglelefteq$ (and its derived notations) is used for dealing with both $\trianglelefteq_s$ and $\trianglelefteq_w$ (and their derived notations).

5.1 General definitions

The *component-graph* $\mathfrak{G}$ of an image $I : \Omega \to V$ is defined as the Hasse diagram of the ordered set $(\Theta, \trianglelefteq)$. However, three variants of component-graphs can be relevantly considered by defining two additional subsets $\dot{\Theta}, \ddot{\Theta} \subseteq \Theta$ of valued connected components

$$\dot{\Theta} = \bigcup_{X \in \Psi} \{X\} \times \overset{\leqslant}{\bigtriangledown} \{v \mid X \in \mathcal{C}[\lambda_v(I)]\} \tag{16}$$

$$\ddot{\Theta} = \bigcap \Big\{ \Theta' \subseteq \Theta \mid I = \overset{\leq}{\bigvee_{K \in \Theta'}} C_K \Big\} \tag{17}$$

Broadly speaking, Θ gathers all the valued connected components induced by I; $\dot{\Theta}$ gathers the valued connected components of maximal values for any connected components; and $\ddot{\Theta}$ gathers the valued connected components associated to the cylinder functions which are sup-generators of I (Equation (3)). (A discussion and some illustrations of these three families of nodes can be found in [9].)

Remark 1 *The definition of Θ, $\dot{\Theta}$ and $\ddot{\Theta}$ is independent from the choice between strong $\trianglelefteq_s$ and weak order $\trianglelefteq_w$.*

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 $\ddot{\Theta}$). From these definitions, we have

$$\ddot{\Theta} \subseteq \dot{\Theta} \subseteq \Theta \tag{18}$$

and

$$\overset{\trianglelefteq}{\bigvee} \Theta = \overset{\trianglelefteq}{\bigvee} \dot{\Theta} = \overset{\trianglelefteq}{\bigvee} \ddot{\Theta} = (\Omega, \bot) \tag{19}$$

$$\overset{\trianglelefteq}{\bigtriangleup} \Theta = \overset{\trianglelefteq}{\bigtriangleup} \dot{\Theta} = \overset{\trianglelefteq}{\bigtriangleup} \ddot{\Theta} \tag{20}$$

Remark 2 *Equations (18–20) are valid for both strong $\trianglelefteq_s$ and weak order $\trianglelefteq_w$.*

We have the following definition for the three variants of component-graphs.

Definition 3 (Component-graph(s)) *Let $I : \Omega \to V$ be an image. 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 to unify the three kinds of component-graphs. The elements of $\mathring{\Theta}$ are called $\mathring{\Theta}$-*nodes *(or simply,* nodes*); the elements of $\mathring{\blacktriangleleft}$ are called $\mathring{\Theta}$-*edges *(or simply,* edges*); $(\Omega, \bot)$ is called the* root*; the elements of $\bigtriangleup^{\trianglelefteq} \mathring{\Theta}$ are called the* leaves *of the $\mathring{\Theta}$-component-graph.*

5.2 Strong versus weak component-graphs

When a component-graph is defined from the strong (resp. weak) order relation $\trianglelefteq_s$ (resp. $\trianglelefteq_w$), it is called a *strong* (resp. *weak*) *component-graph*, and it is noted $\mathring{\mathfrak{G}}_s$ (resp. $\mathring{\mathfrak{G}}_w$).

Figures 5 and 6 illustrate the strong and weak component-graphs $\mathfrak{G}_s$ and $\mathfrak{G}_w$ of the image I depicted in Figure 3.

In the previous literature on this topic (and in particular, in [9]), the only considered component-graphs were the weak ones. In this section, we discuss about the relevance of strong versus weak component-graphs. This discussion will lead us to conclude that the first family is indeed to be preferred to the second, due to its more regular structural properties, while both families have globally the same image modelling abilities.

First, it was observed in Section 5.1 that strong and weak component-graphs rely on the same sets of nodes. In particular, $\mathring{\mathfrak{G}}_s$ and $\mathring{\mathfrak{G}}_w$ have the same root, the same leaves, and the same subfamilies of nodes Θ, $\dot{\Theta}$ and $\ddot{\Theta}$.

Second, both strong and weak component-graphs are relevant generalizations of the notion of component-tree. On the one hand, when $\leqslant$ is a total order, both notions are the same. On the other hand, the component-graph is —such as the component-tree— a lossless image model with respect to the (de)composition formula of Equation (3). This is formalized in the following two propositions.

Proposition 1 *Let $I : \Omega \to V$ be a grey-level image, i.e. $\leqslant$ is a total order on V. The component-tree $\mathfrak{T}$ of I is isomorphic to its (strong and weak) component-graphs $\dot{\mathfrak{G}}$ and $\ddot{\mathfrak{G}}$.*

Remark 3 *This isomorphism does not hold for $\mathfrak{G}$, in general, since $\mathfrak{T}$ gathers the same connected components obtained at successive values as a single node, contrary to $\mathfrak{G}$.*

Proposition 2 *Let $I : \Omega \to V$ be an image. We have*

$$I = \overset{\leq}{\bigvee_{K \in \dot{\Theta}}} C_K = \overset{\leq}{\bigvee_{v \in V}} \overset{\leq}{\bigvee_{X \in \mathcal{C}[\lambda_v(I)]}} C_{(X,v)} \tag{21}$$

Remark 4 *The proof of Propositions 1 and 2 given in [9, Property 18] in the case of weak component-graphs is also valid for strong component-graphs.*

At this point, strong and weak component-graphs present the same properties. Their main difference, discussed hereinafter, actually lies in their respective structure with respect to the Hasse diagram $(V, \prec)$ of the partially ordered set $(V, \leqslant)$ of values.

As already observed in [9], the component-graph $\mathfrak{G}$ locally inherits from the structure of $(V, \prec)$. This is first stated by the following property.

Property 2 ([9]) *Let $K : (X, v) \in \bigtriangleup^{\trianglelefteq} \Theta$. The function*

$$\left| \begin{array}{ll} \sigma : K^{\uparrow} & \to v^{\downarrow} \\ \quad (Y, w) & \mapsto w \end{array} \right. \tag{22}$$

is a bijection between $K^{\uparrow}$ and $v^{\downarrow}$.

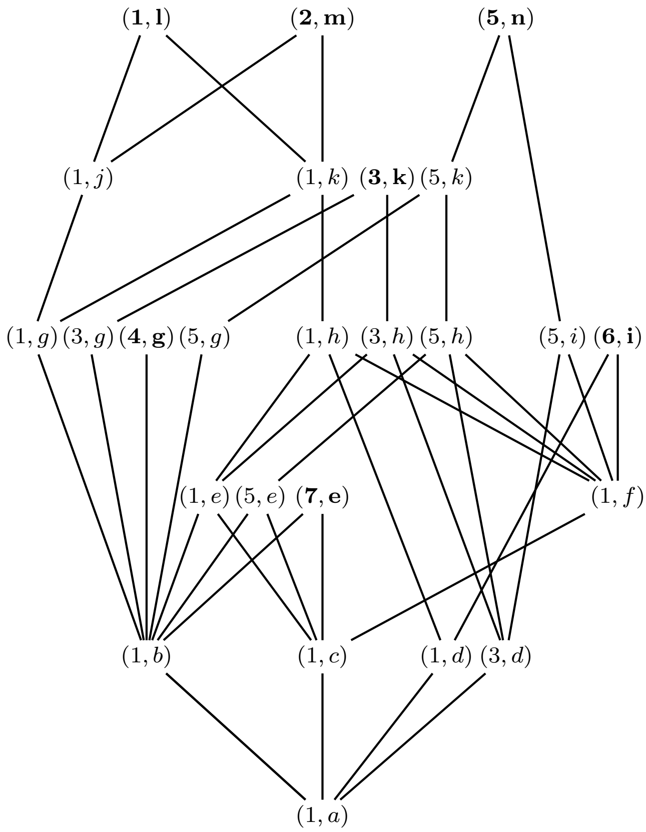

Fig. 5 The strong component-graph $\mathfrak{G}_s$ of the image I of Figure 3. Each couple (ℓ, v) corresponds to a given node / valued connected component of value v, obtained as a connected component of the image I thresholded at v; for instance, $(1, h)$, $(3, h)$ and $(5, h)$ correspond to the three connected components of the subgraph $(\lambda_h(I), \frown)$ of Figure 4. The meaning of the labels ℓ will be explained in Section 10; at this stage, labels are only used to differentiate the nodes. The nodes in bold correspond to leaves of the component-graph. We can observe that for each node, the part of the component-graph located below has exactly the same structure as the corresponding part of the Hasse diagram of $(V, \leqslant)$ (Equation (24)). Note that, contrary to the convention adopted in Figure 2, the higher the nodes, the lower with respect to the order $\trianglelefteq_s$. This reverse choice will allow us to better visualize the links between the component-graphs and the Hasse diagram of $(V, \leqslant)$. In particular, here, the maximum node $(1, a)$ (that corresponds to $(\Omega, \perp)$) is located at the lowest position.

The difference then appears when observing the relationships between $\leqslant$, $\trianglelefteq_s$ and $\trianglelefteq_w$ within the subgraphs associated to the leaves of the strong and weak component-graphs.

Property 3 ([9]) *Let $K = (X, v) \in \Delta^{\trianglelefteq}\Theta$. The function $\sigma^{-1} : v^{\downarrow} \to K^{\uparrow}$ induces a homomorphism from $(v^{\downarrow}, \geqslant)$ to $(K^{\uparrow}, \trianglelefteq_w)$, i.e. for any $K_1 = \sigma^{-1}(v_1)$, $K_2 = \sigma^{-1}(v_2)$, we have*

$$v_1 \geqslant v_2 \Rightarrow K_1 \trianglelefteq_w K_2 \quad (23)$$

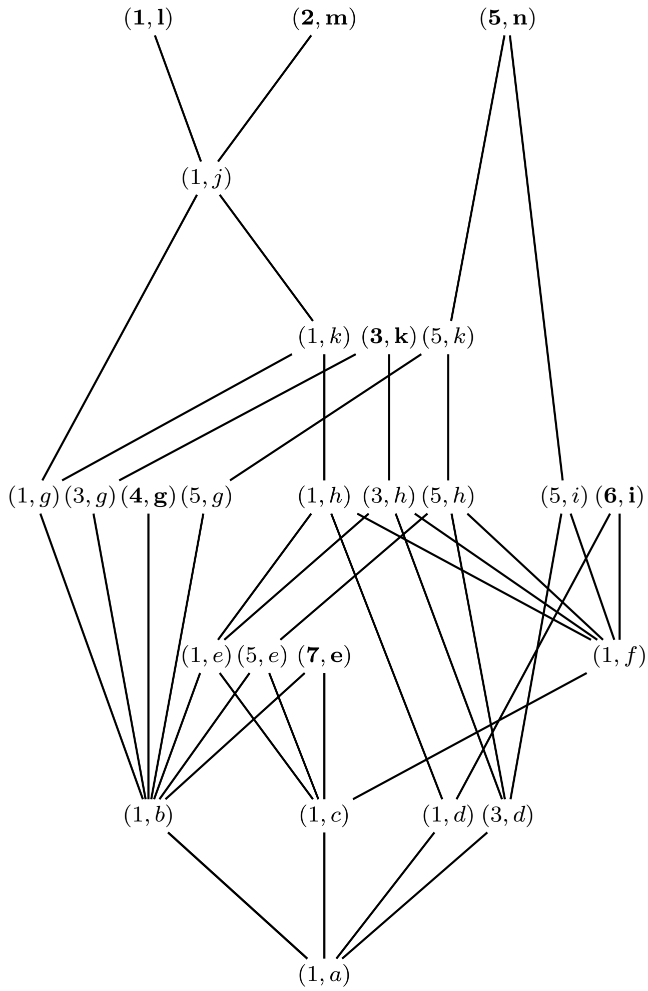

Fig. 6 The weak component-graph $\mathfrak{G}_w$ of the image I of Figure 3. By contrast with the strong component-graph, for each node, the part of the component-graph located below does not necessarily have the same structure as the corresponding part of the Hasse diagram of $(V, \leqslant)$ (Equation (23)). This is, for instance, the case below the nodes $(1, l)$ and $(2, m)$ where the nodes $(1, j)$ and $(1, k)$ are comparable with respect to $\trianglelefteq_w$ due to the inclusion of their supports, whereas their values j and k are not comparable with respect to $\leqslant$.

Property 4 *Let $K = (X, v) \in \Delta^{\trianglelefteq}\Theta$. The function $\sigma^{-1} : v^{\downarrow} \to K^{\uparrow}$ induces an isomorphism between $(v^{\downarrow}, \geqslant)$ and $(K^{\uparrow}, \trianglelefteq_s)$, i.e. for any $K_1 = \sigma^{-1}(v_1)$, $K_2 = \sigma^{-1}(v_2)$, we have*

$$v_1 \geqslant v_2 \Leftrightarrow K_1 \trianglelefteq_s K_2 \quad (24)$$

Proof We set $K_1 = (X_1, v_1)$ and $K_2 = (X_2, v_2)$. The "$\Rightarrow$" part of Equation (24) (and also Equation (23)) derives from the fact that $X_1 \cap X_2 \neq \emptyset$. Then, $v_2 \leqslant v_1$ implies $X_1 \subseteq X_2$, and it follows that $K_1 \trianglelefteq_s K_2$ (and a fortiori, $K_1 \trianglelefteq_w K_2$). The "$\Leftarrow$" part of Equation (24) derives from the very definition of $\trianglelefteq_s$. $\square$

In other words, the part of a component-graph located between the root $(\Omega, \perp)$ and each leaf $K = (X, v)$ contains some nodes that are directly associated to the values of V located between $\perp$ and v. In the case of the strong component-graph, the structure of the Hasse diagram of $(K^{\downarrow}, \trianglelefteq_s)$ is exactly the same as the structure of the Hasse diagram of $(v^{\uparrow}, \geqslant)$. By contrast, in the case of the weak component-graph, this structure is not always the same, and depends in priority on the $\subseteq$ relation between the supports of the nodes.

Since both strong and weak component-graphs rely on the same nodes, and are defined from order relations that are relevant (they depend on both $\subseteq$ and $\leqslant$) and compliant with the component-tree in the case of grey-level images, it is preferable to focus on the strong component-graph, that presents a closer similarity with the Hasse diagram of $(V, \leqslant)$. This will allow us to take advantage of this known data-structure during the construction process.

Before concluding this structural study, it is worth mentioning that the strong and weak component-graphs are the same under specific hypotheses.

Property 5 *Let $K, K' \in \ddot{\Theta}$. We have*

$$K \ddot{\blacktriangleleft}_s K' \Leftrightarrow K \ddot{\blacktriangleleft}_w K' \tag{25}$$

i.e., we have $\ddot{\mathfrak{G}}_s = \ddot{\mathfrak{G}}_w$.

Proof Let $K = (X, v), K' = (Y, w) \in \ddot{\Theta}$. Let us suppose that $K \trianglelefteq_w K'$. Then, we have either (1) $X \subset Y$ or (2) $X = Y$ and $w \leqslant v$. In case (2), we directly have $K \trianglelefteq_s K'$. Now, let us consider that case (1) holds. Since $K \in \ddot{\Theta}$, there exists $x \in X$ such that $I(x) = v$. Since $X \subset Y$, we have $x \in Y$. But $Y \subseteq \lambda_w(I)$ and then $w \leqslant v$. It follows that $K \trianglelefteq_s K'$. We then have $K \trianglelefteq_w K' \Rightarrow K' \trianglelefteq_w K$, and from Property 1, it comes that $K \trianglelefteq_w K' \Leftrightarrow K' \trianglelefteq_w K$. The result follows by transitive reduction of $\trianglelefteq_w$ and $\trianglelefteq_s$. $\square$

Property 6 *Let us suppose that $(V, \leqslant)$ is a lattice*[1]*. Let $K, K' \in \dot{\Theta}$. We have*

$$K \dot{\blacktriangleleft}_s K' \Leftrightarrow K \dot{\blacktriangleleft}_w K' \tag{26}$$

i.e., for lattices $(V, \leqslant)$, we have $\dot{\mathfrak{G}}_s = \dot{\mathfrak{G}}_w$.

Proof Let $K = (X, v), K' = (Y, w) \in \dot{\Theta}$. Let us suppose that $K \trianglelefteq_w K'$. Then, we have either (1) $X \subset Y$ or (2) $X = Y$ and $w \leqslant v$. In case (2), we directly have $K \trianglelefteq_s K'$. Now, let us consider that case (1) holds. Let $u = \bigvee^{\leqslant}\{v, w\} \in V$. Let us consider $K'' = (Z, u)$ such that $K'' \trianglelefteq_w K$ and $K'' \trianglelefteq_w K'$. Such a node necessarily exists. Let $x \in X$. Then, we have $x \in \lambda_v(I)$, and thus $v \leqslant I(x)$. Since $x \in X \subset Y$, we have $x \in \lambda_w(I)$, and thus $w \leqslant I(x)$. But then, we have $u = \bigvee^{\leqslant}\{v, w\} \leqslant I(x)$. It follows that $X \subseteq Z$, and since $K'' \trianglelefteq_w K$, we also have $Z \subseteq X$, and then $X = Z$. It comes from the definition of $\dot{\Theta}$ (Equation (16)) that $K'' = K$, and thus $u = v$. It follows that $w \leqslant \bigvee^{\leqslant}\{v, w\} = u = v$. Then we have $K \trianglelefteq_s K'$. Consequently, we have $K \trianglelefteq_w K' \Rightarrow K' \trianglelefteq_s K$, and from Property 1, it comes that $K \trianglelefteq_w K' \Leftrightarrow K' \trianglelefteq_w K$. The result follows by transitive reduction of $\trianglelefteq_w$ and $\trianglelefteq_s$. $\square$

[1] This means that for all $v, w \in V$, the two elements $\bigvee^{\leqslant}\{v, w\}$ and $\bigwedge^{\leqslant}\{v, w\}$ exist and belong to V.

In the next sections, we describe how to build the (strong) component-graph of an image.

6 Flat zones and image reduction

Let $I : \Omega \to V$ be an image defined on the graph $(\Omega, \frown)$, and taking its values in the ordered set $(V, \leqslant)$. Our purpose is to build the strong component-graph $\mathfrak{G}_s$ of I.

As a preliminary step, we show hereafter that a (weak or strong) component-graph is isomorphic to the component-graph of the "flat zone" image [18] associated to I. This result will be useful for reducing the space and time cost of the algorithmic process of component-graph construction.

Let $v \in V$. Let $\frown_v$ be the relation defined on Ω as follows:

$$x \frown_v y \Leftrightarrow ((x \frown y) \wedge (I(x) = I(y))) \tag{27}$$

Let $\leftrightarrow_v$ be the reflexive-transitive closure $\frown_v$.

We define the relation $\leftrightarrow_V$ as the union of the $\leftrightarrow_v$, for all $v \in V$

$$x \leftrightarrow_V y \Leftrightarrow (\exists v \in V, x \leftrightarrow_v y) \tag{28}$$

The relation $\leftrightarrow_V$ is an equivalence relation, that induces a partition of Ω with respect to the maximal connected sets of same values. These sets are called the *flat zones.*

Definition 4 (Flat zone) *Let $I : \Omega \to V$ be an image. The set of the* flat zones *of I, noted Φ is the partition of Ω defined by $\Phi = \Omega/\leftrightarrow_V$.*

Let us consider the relation $\frown_\Phi$ defined on Φ, for any distinct $X, Y \in \Phi$, as follows:

$$X \frown_\Phi Y \Leftrightarrow (\exists x \in X, y \in Y, x \frown y) \tag{29}$$

This relation $\frown_\Phi$ is the adjacency (irreflexive, symmetric) relation on the flat zones of I induced by the adjacency relation on Ω.

From the graph $(\Phi, \frown_\Phi)$, we can define the flat zone image I_Φ induced by I as follows

$$\left| \begin{array}{lll} I_\Phi : \Phi & \to V \\ \quad [x]_{\leftrightarrow_V} & \mapsto I(x) \end{array} \right. \tag{30}$$

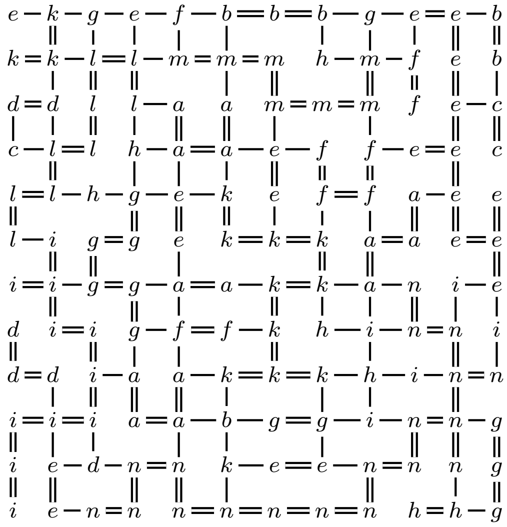

Fig. 7 The flat zone image I_Φ associated to the image I of Figure 3. The maximal connected sets of same values linked by double edges —namely, the flat zones— correspond to the connected components for the equivalence relation $\leftrightarrow_V$. Each of these sets of Ω is then an element of Φ. The adjacency relation $\frown_\Phi$ between two elements of Φ is depicted by a single edge between them. This adjacency $\frown_\Phi$ is inherited from the adjacency $\frown$ on Ω (Equation (29)).

Figure 7 illustrates the flat zone image I_Φ associated to the image I of Figure 3.

We note $\Theta_\Phi = \bigcup_{v \in V} \mathcal{C}[\lambda_v(I_\Phi)] \times \{v\}$ the set of the valued connected components of I_Φ. The following property is a direct consequence of the definitions of Φ and $\frown_\Phi$.

Property 7 *The function*

$$\left| \begin{array}{ll} \phi : \Theta & \to \Theta_\Phi \\ \quad (X, v) & \mapsto (X/\!\leftrightarrow_V, v) \end{array} \right. \tag{31}$$

is a bijection between Θ and Θ_Φ. Its inverse function is

$$\left| \begin{array}{ll} \phi^{-1} : \Theta_\Phi & \to \Theta \\ \quad (A, v) & \mapsto (\bigcup_{X \in A} X, v) \end{array} \right. \tag{32}$$

Then, the following proposition establishes that we can work indistinctly on an image I or its flat zone analogue I_Φ, in order to build and use a component-graph.

Proposition 3 *The bijection ϕ induces an isomorphism between the component-graphs $\mathring{\mathfrak{G}} = (\mathring{\Theta}, \mathring{\blacktriangleleft})$ of I and $\mathring{\mathfrak{G}}_\Phi = (\mathring{\Theta}_\Phi, \mathring{\blacktriangleleft}_\Phi)$ of I_Φ. More precisely, we have*

$$((X, v) \mathring{\blacktriangleleft} (Y, w)) \Leftrightarrow ((\phi(X), v) \mathring{\blacktriangleleft}_\Phi (\phi(Y), w)) \tag{33}$$

Proof Let $X, Y \in \Theta$. From the definition of ϕ, we have $X \subseteq Y \Leftrightarrow \phi(X) \subseteq \phi(Y)$. Then, it follows from the definition of $\trianglelefteq_w$ and $\trianglelefteq_s$ (simply noted $\trianglelefteq$ hereinafter), that $((X, v) \trianglelefteq (Y, w)) \Leftrightarrow ((\phi(X), v) \trianglelefteq_\Phi (\phi(Y), w))$. This is a fortiori true for the transitive reductions $\blacktriangleleft$ and $\blacktriangleleft_\Phi$ of $\trianglelefteq$ and $\trianglelefteq_\Phi$, respectively. Still from the definition of ϕ, we have $\phi(\ddot{\Theta}) = \ddot{\Theta}_\Phi$, and finally, the isomorphism between $(\Theta, \trianglelefteq)$ and $(\Theta_\Phi, \trianglelefteq_\Phi)$ also implies that $\phi(\dot{\Theta}) = \dot{\Theta}_\Phi$. As a consequence, Equation (33) holds in any cases. □

Property 8 *The computation of $(\Phi, \frown_\Phi)$ from $(\Omega, \frown)$ can be done in linear time $\mathcal{O}(|\frown|)$.*

Practically, it is then relevant to work with the flat zone image I_Φ, where each flat zone of I —composed of $k \geq 1$ points— is replaced by a unique point. Doing so, the space complexity of the input image I is reduced from $|\Omega|$ to $|\Phi|$, with $|\Phi| \leq |\Omega|$ (and $|\frown| \leq |\frown_\Phi|$), and in the most favourable cases $|\Phi| \ll |\Omega|$ (and $|\frown| \ll |\frown_\Phi|$). This reduction of space complexity will also have an impact on the time complexity of the different steps of the component-graphs construction process.

Of course, from Equations (21) and (30), it follows that the information carried by the component-graph of I_Φ is the same as that carried by I. Any image processing operation based on $\mathring{\mathfrak{G}}$ can then be performed the same way on $\mathring{\mathfrak{G}}_\Phi$.

Remark 5 *From now on, we will no longer consider the image I, but its flat zone image I_Φ. For the sake of readability, we will still note I instead of I_Φ, and Ω instead of Φ. In particular, we will assume that for a given (flat zone) image I, we have for any $x, y \in \Omega$*

$$(x \frown y) \Rightarrow (I(x) \neq I(y)) \tag{34}$$

Remark 6 *In digital imaging, we have $|\frown| = \mathcal{O}(|\Omega|)$. For instance, with 4-, 8-, 6- and 26-adjacencies on $\mathbb{Z}^2$ and $\mathbb{Z}^3$, we have $|\frown| = k.|\Omega|$, with $k = 2, 4, 3$ and 13, respectively. This generally still holds for the induced flat zone images. For the sake of concision, we will assume from now on that we have $|\frown| = \mathcal{O}(|\Omega|)$.*

7 Building the leaves of a component-graph

We are now ready to start building a (strong) component-graph. The cornerstone of this algorithmic process is the notion of a leaf.

A leaf L is a minimal element of $\mathring{\mathfrak{G}}$, i.e. $L \in \triangle^{\trianglelefteq} \mathring{\Theta}$ (Definition 3 and Equation (20)).

Since Ω is finite and $\leqslant$ is antisymmetric, there exist one or many points $x \in \Omega$ such that for all $y \frown x$, we have $I(x) \not< I(y)$, i.e. $I(x) \in V$ is a locally maximal value of the image I. Then, it is plain that $L_x = (\{x\}, I(x))$ is a node of $\mathring{\mathfrak{G}}$, i.e. $L_x \in \mathring{\Theta}$.

The characterization of a leaf relies on a local criterion: it is sufficient to observe the values of the adjacent

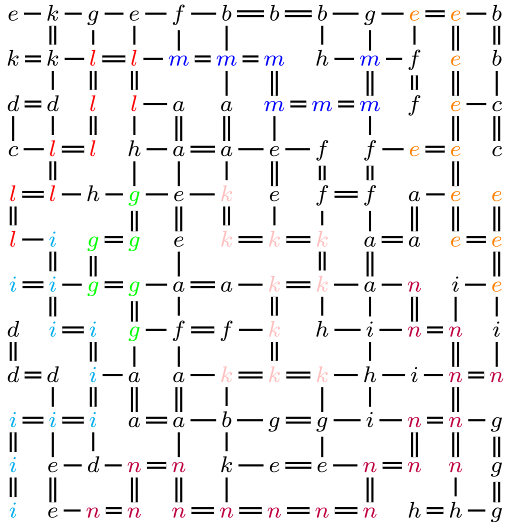

Fig. 8 The 7 leaves of the image I of Figure 7 are depicted in colour. From now on, I_Φ will be noted as I for the sake of concision. In particular, a flat zone is considered as a single element of Ω (formerly, Φ). For instance, the blue flat zone composed of 7 points $x \in \Omega$ of value $I(x) = m$ is now a single point, and it is adjacent to 9 other elements of (lower or non-comparable) values l, f, b, h, g, f, f, e and a. However, we keep the initial Cartesian grid formalism for the sake of readability and for allowing comparisons with Figures 3 and 7.

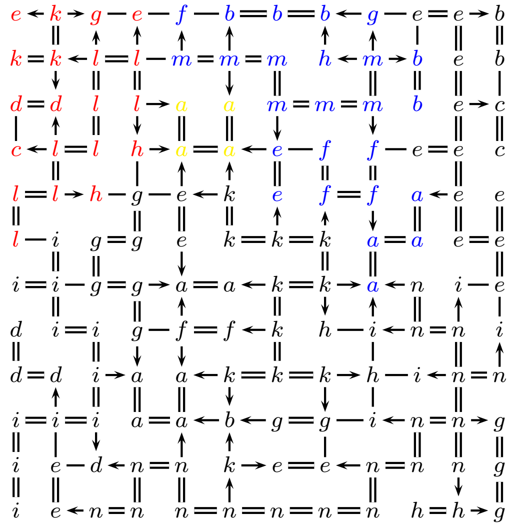

Fig. 9 Two reachable zones of the image I of Figures 3 and 7. Here, an arrow pointing from an element $x \in \Omega$ of value $I(x) \in V$ to an element $y \in \Omega$ of value $I(y)$ indicates that x and y are adjacent (i.e. $x \frown y$) and $I(x) > I(y)$ (a segment with no arrow indicates that $x \frown y$ whereas $I(x)$ and $I(y)$ are non-comparable). A first reachable zone $\rho(x_1)$ is depicted in red and yellow for the leaf-point $x_1 \in \Lambda$ of value $I(x_1) = l$. A second reachable zone $\rho(x_2)$ is depicted in blue and yellow for the leaf-point $x_2 \in \Lambda$ of value $I(x_2) = m$. The overlapping part between $\rho(x_1)$ and $\rho(x_2)$ is depicted in yellow; here, it corresponds to only one point of value a.

points of a candidate point of I. Then, we can compute all the leaves of $\mathring{\mathfrak{G}}$ by an exhaustive scanning of $(\Omega, \frown)$, with a linear time cost $\mathcal{O}(|\Omega|)$.

Figure 8 illustrates the leaves of the image I of Figure 7.

In the sequel, we will denote by $\Lambda \subseteq \Omega$ the set of all the points of Ω that correspond to supports of leaves; these points will be called *leaf-points*. In other words, we have

$$\overset{\trianglelefteq}{\triangle} \mathring{\Theta} = \{L_x = (\{x\}, I(x)) \mid x \in \Lambda\} \tag{35}$$

The notion of a leaf is fundamental for understanding the structure of a component-graph, and then for building it. In particular, in the next section, we show that the support Ω of the image I can be decomposed into regions, each region being associated to a specific leaf / leaf-point. From these regions, and more precisely the adjacency relation between them, it will be possible to efficiently build the strong component-graph.

8 Influence zones of an image

Each point $x \in \Omega$ has a given value $I(x) \in V$. It is also adjacent to other points $y \in \Omega$ (i.e. $x \frown y$) with values $I(y)$. From Equation (34), we have either $I(x) < I(y)$ or $I(x) > I(y)$ or $I(x), I(y)$ non-comparable.

From a leaf-point $x \in \Lambda$ corresponding to a leaf L_x, we can reach certain points $y \in \Omega$ by a descent paradigm. More precisely, for such points y, there exists a sequence $x = x_0 \frown \ldots \frown x_i \frown \ldots \frown x_t = y$ $(t \geq 0)$ in Ω such that for any $i \in [\![0, t-1]\!]$, we have $I(x_i) > I(x_{i+1})$. In such case, we note $x \searrow^{\Omega} y$. This leads to the following notion of a *reachable zone*.

Definition 5 (Reachable zone) *Let $x \in \Lambda$ be a leaf-point of I. The* reachable zone *of x (in Ω) is the set*

$$\rho(x) = \{y \in \Omega \mid x \searrow^{\Omega} y\} \tag{36}$$

We note $P = \{\rho(x) \mid x \in \Lambda\}$ the set of all the reachable zones of I.

Property 9 *The set P is a cover of Ω.*

The important fact of this property is that $\bigcup_{x \in \Lambda} \rho(x) = \Omega$. However, the set of reachable zones is not a partition of Ω, in general, due to possible overlappings. For instance, let $x_1, x_2 \in \Lambda$, $y \in \Omega$, and let us suppose that $x_1 \frown y \frown x_2$ and $x_1 > y < x_2$; then we have $y \in \rho(x_1) \cap \rho(x_2) \neq \emptyset$. An example of two (overlapping) reachable zones is also exemplified in Figure 9.

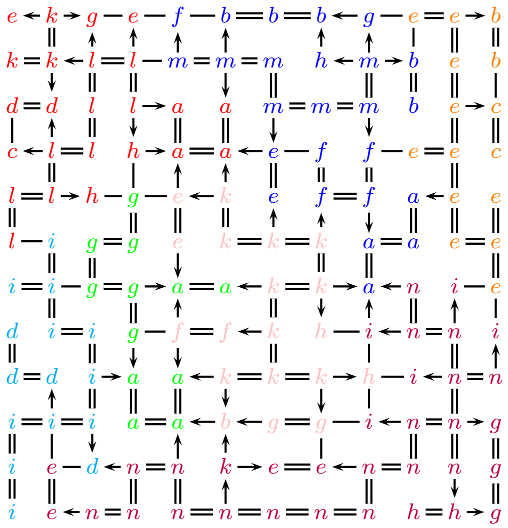

Fig. 10 A partition Σ of influence zones for the image I of Figures 3 and 7. The 7 colours correspond to the 7 influence zones, where each influence zone is associated to one leaf / leaf-point (Figure 8).

Remark 7 *The computation of P can be carried out by a seeded region-growing process, with Λ as set of seeds. The time cost is output-dependent, since it is linear with respect to the ratio of overlap between the different reachable zones. More precisely, it is $\mathcal{O}(\sum_{x \in \Lambda} |\rho(x)|) = \mathcal{O}((1+\gamma).|\Omega|)$, with $\gamma \in [0, |\Omega|/4] \subset \mathbb{R}$ the overlap ratio, varying between 0 (no overlap between reachable zones, i.e. $\{\rho(x) \mid x \in \Lambda\}$ is a partition of Ω) and $|\Omega|/4$ (all reachable zones are maximally overlapped). The upper bound of γ is $(|\Omega|-1)^2/4|\Omega|$, and is reached when $|\Lambda| = (|\Omega|+1)/2$.*

Practically, in the worst cases, the time cost for computing P is quadratic, namely $\mathcal{O}(|\Omega|^2)$. However, from an algorithmic point of view, we need not computing P as a whole. Actually, we only need a partition Σ of Ω that satisfies the following properties. This leads us to define a second notion of an *influence zone*.

Definition 6 (Influence zone) *A set of influence zones of I, noted Σ, is a partition of Ω defined such that for any $S \in \Sigma$, we have:*

(i) $|\Lambda \cap S| = 1$
(ii) $x \in \Lambda \cap S \Rightarrow S \subseteq \rho(x)$
(iii) $\forall y \in S, x \searrow^S y$

In other words, there exists a unique leaf-point $x \in \Lambda \cap S$ (i) and the set S is included in the influence zone $\rho(x)$ of this unique leaf-point; in addition, any point y of S can be reached from x by a descending path in S. For any leaf-point $x \in \Lambda$, the set $\sigma(x) \in \Sigma$ such that $x \in \sigma(x) \subseteq \rho(x)$ is called an influence zone of x (in I).

Remark 8 *Contrary to the cover P of reachable zones, a partition Σ of influence zones is not unique, in general, for a given image I. However, this non-determinism corresponds to overlapping areas in reachable zones (namely, values accessible by a descent paradigm from various leaves), and then it will not have any impact on the component-graph construction result. The influence zones can be computed by a concurrent seeded region-growing process, considering Λ as set of seeds. The time cost for computing the partition Σ is no longer output-dependent, and is linear, namely $\mathcal{O}(|\Omega|)$. An example of partition Σ is illustrated in Figure 10.*

9 Influence zone graph of an image

The partition Σ subdivides the support Ω of the image I into $|\Lambda|$ regions ($1 \leq |\Lambda| \leq |\Omega|$). Each of these regions $\sigma(x)$ contains exactly one leaf-point $x \in \Lambda$ associated to a leaf $L_x = (\{x\}, I(x))$ of the component graph $\mathring{\mathfrak{G}}$.

We are now describing the structure of these $|\Lambda|$ regions within the image support. In particular, we are interested by the adjacency relations between them. Indeed, this will allow us to further compute the nodes and their relations in the strong component-graph.

To this end, we define a graph called the *influence zone graph*, where vertices are the regions $\sigma(x)$, $x \in \Lambda$, and the adjacency relation is induced by the adjacency relation $\frown$ on Ω. Since the function $\sigma : \Lambda \to \Sigma$ is a bijection, we can define, without loss of generality, this adjacency relation —then noted $\frown_\Lambda$— on Λ instead of Σ.

Definition 7 (Influence zone graph) *The* influence zone graph *of an image I is the graph $\mathfrak{Z} = (\Lambda, \frown_\Lambda)$ where $\frown_\Lambda$ is defined, for any distinct leaf-points $x, y \in \Lambda$ as*

$$(x \frown_\Lambda y) \Leftrightarrow (\exists (x', y') \in \sigma(x) \times \sigma(y), x' \frown y') \tag{37}$$

Figure 11 illustrates the influence zone graph of the partition Σ of I defined in Figure 10.

Let us consider two distinct leaf-points $x, y \in \Lambda$ such that $x \frown_\Lambda y$. There exists a path $x = x_0 \frown \ldots \frown x_i = x' \frown y' = x_{i+1} \frown \ldots \frown x_t = y$ $(0 \leq i < t \geq 1)$ in Ω, such that $S_x = \{x_j\}_{j=0}^{i} \subseteq \sigma(x)$ and $S_y = \{x_j\}_{j=i+1}^{t} \subseteq \sigma(y)$.

By construction, for all $p \in S_x$ (resp. S_y), we have $I(p) \geqslant I(x')$ (resp. $I(p) \geqslant I(y')$). Then, for all $p \in S_x \cup S_y$, and for all $v \in I(x')^{\downarrow} \cap I(y')^{\downarrow}$, we have $I(p) \geqslant v$.

This leads us to define a valuation of the edges of $\frown_\Lambda$ as follows

$$\left| \begin{array}{ll} \nu : \frown_\Lambda & \to 2^V \\ \quad (x,y) & \mapsto \bigcup_{(x',y') \in E(x,y)} I(x')^{\downarrow} \cap I(y')^{\downarrow} \end{array} \right. \tag{38}$$

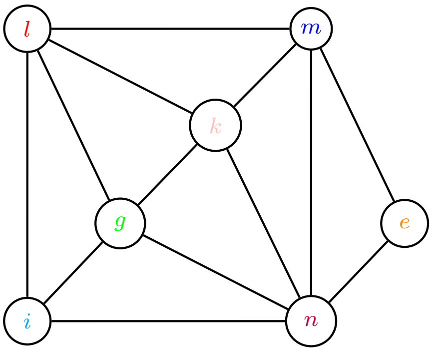

Fig. 11 The influence zone graph $\mathfrak{Z} = (\Lambda, \frown_\Lambda)$ associated to the partition Σ (Figure 10) of the image I of Figures 3 and 7. Each disk corresponds to a leaf-point $x \in \Lambda$. The value in the disk is the value $I(x)$ of this leaf-point, and the color indicates the corresponding influence zone in Figure 10. The adjacency $\frown_\Lambda$ is represented by segments between the disks.

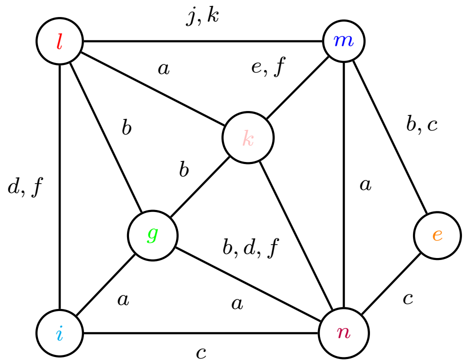

Fig. 12 The influence zone graph $\mathfrak{Z} = (\Lambda, \frown_\Lambda)$ of Figure 11 is now endowed with the function ν^∇. For each pair of adjacent leaf-points $x \frown_\Lambda y$, the set of values $\nu^\nabla((x, y))$ is provided as a valuation of the edge between the two related vertices of $\mathfrak{Z}$. The computation of these sets of values first requires to compute the sets $E(x, y)$ (Equation (39)), and then $\nu^\nabla((x, y))$ by following Equation (41). For instance, for the red (l) and blue (m) leaf-points, noted x_1 and x_2, respectively, the set $E(x_1, x_2)$ is composed of four pairs of points, that correspond to the four edges in Figure 10 linking a red point x' and a blue point y'. The associated four pairs of values $(I(x'), I(y'))$ are (e, f), (l, m), (a, m) and (a, e). Then the four sets $\nabla^\leqslant I(x')^\downarrow \cap I(y')^\downarrow$ are $\{c\}$, $\{j, k\}$, $\{a\}$ and $\{a\}$. The union of these four sets is $\bigcup_{(x',y') \in E(x_1,x_2)} \nabla^\leqslant I(x')^\downarrow \cap I(y')^\downarrow = \{a, c, j, k\}$ and the set of maximal elements is $\nabla^\leqslant \bigcup_{(x',y') \in E(x_1,x_2)} \nabla^\leqslant I(x')^\downarrow \cap I(y')^\downarrow = \{j, k\}$ (Equation (41)). We then have $\nu^\nabla((x_1, x_2)) = \{j, k\}$.

where

$$E(x, y) = \{(x', y') \mid \sigma(x) \ni x' \frown y' \in \sigma(y)\} \tag{39}$$

Th function ν provides the values v of V that allow us to define sequences of points between two adjacent influence zones, that connect the two associated leaf-points, while remaining below v.

More formally, we have the following property.

Property 10 *Let $x, y \in \Lambda$ such that $x \frown_\Lambda y$. The following two statements are equivalent*

(i) $v \in \nu((x, y))$*; and*
(ii) there exists a sequence $x = x_0 \frown \ldots \frown x_t = y$ $(t \geq 1)$ *such that for all* $i \in [\![0, t]\!]$, $x_i \in \sigma(x) \cup \sigma(y)$ *and* $I(x_i) \geqslant v$.

Remark 9 *It is sufficient to consider the restricted function* $\nu^\nabla : \frown_\Lambda \to 2^V$ *defined, for any* $x \frown_\Lambda y$ *as*

$$\nu^\nabla((x, y)) = \overset{\leqslant}{\nabla} \nu((x, y)) \tag{40}$$

$$= \overset{\leqslant}{\nabla} \bigcup_{(x',y') \in E(x,y)} \overset{\leqslant}{\nabla} I(x')^\downarrow \cap I(y')^\downarrow \tag{41}$$

(In practice, the formulation of Equation (41) is used for actually computing $\nu^\nabla((x, y))$*.) Using* ν^∇ *allows us to reduce the space cost of* ν*, by only storing the minimal set of values required to identify the paths between the leaf-points* x *and* y *within* $\sigma(x) \cup \sigma(y)$*. In particular, in Property 10, Statement (i) then rewrites as "*$\exists w \in \nu^\nabla((x, y)), v \leqslant w$*" or "*$v \in \bigcup_{w \in \nu^\nabla((x,y))} w^\downarrow$*".*

Remark 10 *From an algorithmic point of view,* ν^∇ *can be built independently for any distinct* $x \frown_\Lambda y$*. In particular, for each* $x \frown_\Lambda y$*, the couples of points* $(x', y') \in E(x, y)$ *can be gathered during the construction of the influence zones* $\sigma(x)$ *and* $\sigma(y)$*, and* ν^∇ *can then be built on the flight.*

Figure 12 provides the function ν^∇ associated to the adjacency $\frown_\Lambda$ of the influence zone graph $\mathfrak{Z} = (\Lambda, \frown_\Lambda)$ of Figure 11.

10 Component-graph modelling

10.1 Node modelling

A node $K = (X, v)$ corresponds to a part $X \subseteq \Omega$ of the support of I, and a value $v \in V$. By construction, there exists (at least) one leaf $L_x = (\{x\}, I(x)) \in \Delta^\trianglelefteq \mathring{\Theta}$ such that $L_x \trianglelefteq K$. In particular, we have $v \leqslant I(x)$ and $x \in X$. In addition, K is the only node of value v such that $x \in X$. As a consequence, the leaf-point x can be used for identifying K. In other words, we can rewrite K as $N(x, v)$, that means "the node of $\mathfrak{G}$ of value v, whose support X contains x".

More formally, we can define the function

$$\left| \begin{array}{l} N : \Lambda \times V \to \Theta \cup \{\alpha\} \\ \quad (x, v) \;\; \mapsto K = (X, v) \text{ with } x \in X \end{array} \right. \tag{42}$$

We set $N(x, v) = \alpha$ whenever $v \not\leqslant I(x)$. This is a convention that allows us to make N fully defined on $\Lambda \times V$

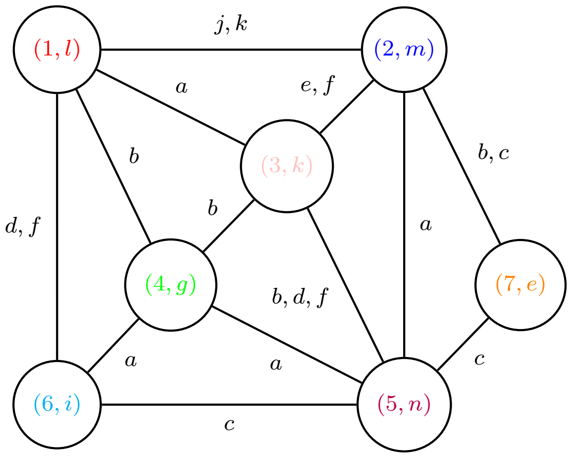

Fig. 13 The influence zone graph $\mathfrak{Z} = (\Lambda, \frown_\Lambda)$ of Figures 11 and 12, where each leaf-point x of value $I(x) = v$ has now a specific label ℓ. Each couple (ℓ, v) then characterizes one of the 7 leaves of the component-graph $\mathfrak{G}$ of the image I of Figures 3 and 7.

despite the fact that all the couples (x, v) are not necessarily associated to a node of the component-graph.

The function $N : \Lambda \times V \to \Theta$ enables to identify all the nodes of the component-graph $\mathfrak{G}$. In other words, N is surjective (note that if $N^{-1}(\{\alpha\}) = \emptyset$, we can omit α in the definition of N, and surjectivity is still preserved). However, a node $K = (X, v)$ may be associated to many couples (x, v), i.e. many leaf-points x can belong to X. In other words, N may be non-injective.

In order to tackle this issue, we assign a numerical label to each leaf-point. We define a bijective function

$$\left| \begin{array}{l} \ell : \Lambda \to [\![1, |\Lambda|]\!] \subset \mathbb{N} \\ \quad x \mapsto \ell(x) \end{array} \right. \tag{43}$$

We note $\ell_x = \ell(x)$ the label of the leaf-point x. If x has a label $\ell(x) = k$, it is also noted x_k.

Then, we can define, for any $K = (X, v) \in \Theta$ a canonical leaf-point x for K, noted x_K. It is defined as the leaf-point $x \in X$ of minimal label, that is

$$x_K = \arg_\Lambda \min\{\ell(x) \mid x \in X\} \tag{44}$$

The label of this canonical leaf-point x_K is noted ℓ_K. The node K is then characterized by the couple (ℓ_K, v).

Remark 11 *A leaf $L_x = (\{x\}, I(x)) \in \Delta^{\trianglelefteq}\Theta$ is necessarily characterized by the couple $(\ell_x, I(x))$.*

In Figure 13, we provide a specific label ℓ for each of the leaf-points of the influence zone graph of Figures 11 and 12. The obtained couples (ℓ, v) then correspond to the actual leaves of the associated component-graph.

Property 11 *For any $K_1, K_2 \in \Theta$, we have*

$$K_1 \trianglelefteq K_2 \Rightarrow \ell_{K_2} \leq \ell_{K_1} \tag{45}$$

In particular we have $\ell_{(\Omega,\bot)} = 1$.

Remark 12 *From now on, we will note indistinctly a node $K = (X, v)$ as K or (ℓ_K, v).*

From a structural point of view, building the set of nodes $\mathring{\Theta}$ of the component-graph $\mathring{\mathfrak{G}}$ is then equivalent to determining the subset of all the couples (ℓ_K, v) within $[\![1, |\Lambda|]\!] \times V$ that characterize these nodes.

10.2 Edge modelling

Until now, the discussions were valid for both strong and weak component-graphs.

At this stage, we now focus on the strong component-graph $\mathring{\mathfrak{G}}_s$, and especially on $\mathfrak{G}_s$, that contains all the nodes of Θ. In the next sections, we will aim at building $\mathfrak{G}_s$.

The strong component-graph $\mathfrak{G}_s$ is the Hasse diagram $(\Theta, \blacktriangleleft_s)$ of the ordered set $(\Theta, \trianglelefteq_s)$. From the very definition of $\trianglelefteq_s$, we have the following property.

Property 12 *Let $K_1 = (X_1, v_1), K_2 = (X_2, v_2) \in \Theta$. We have*

$$K_1 \blacktriangleleft_s K_2 \Rightarrow v_2 \prec v_1 \tag{46}$$

This equation rewrites as

$$(\ell_{K_1}, v_1) \blacktriangleleft_s (\ell_{K_2}, v_2) \Rightarrow v_2 \prec v_1 \tag{47}$$

In other words, an edge of $\mathfrak{G}_s$ between K_1 and K_2 can be modelled by the couple of labels (ℓ_{K_2}, ℓ_{K_1}) associated to the edge (v_2, v_1) of $(V, \prec)$.

10.3 Strong component-graph modelling

We now show how the strong component-graph $\mathfrak{G}_s$ can be modelled (and then stored) by enriching the Hasse diagram $(V, \prec)$.

First, we observed that a node $K = (X, v)$ can be characterized as (ℓ_K, v), where $\ell_K \in [\![1, |\Lambda|]\!] \subset \mathbb{N}$ is the minimal label associated to the canonical leaf-point of K. Consequently, we can define the function

$$\left| \begin{array}{l} \theta : V \to 2^{[\![1, |\Lambda|]\!]} \\ \quad v \mapsto \{\ell_K \mid K = (X, v) \in \Theta\} \end{array} \right. \tag{48}$$

Second, we observed that an edge between two nodes $K_1 = (\ell_{K_1}, v_1)$ and $K_2 = (\ell_{K_2}, v_2)$ can be modelled as the couple (ℓ_{K_2}, ℓ_{K_1}) associated to the edge between v_2 and v_1 in $(V, \prec)$. Consequently, we can define the function

$$\left| \begin{array}{ll} \varepsilon : \prec & \to 2^{[\![1,|\Lambda|]\!] \times [\![1,|\Lambda|]\!]} \\ (v_2, v_1) & \mapsto \{(\ell_{K_2}, \ell_{K_1}) \mid (\ell_{K_1}, v_1) \blacktriangleleft_s (\ell_{K_2}, v_2)\} \end{array} \right. \tag{49}$$

The information carried by the enriched Hasse diagram $(V, \prec, \theta, \varepsilon)$ is sufficient for modelling the strong component-graph $\mathfrak{G}_s = (\Theta, \blacktriangleleft_s)$. This is formalized by the following proposition, that directly derives from the above definitions and properties.

Proposition 4 *Let $\mathfrak{G}_s$ be the strong component-graph of an image I, and $(V, \prec, \theta, \varepsilon)$ the associated enriched Hasse diagram. Then, we have:*

$$K \in \Theta \Leftrightarrow \ell_K \in \theta(v) \tag{50}$$

$$K_1 \blacktriangleleft_s K_2 \Leftrightarrow (\ell_{K_2}, \ell_{K_1}) \in \varepsilon((v_2, v_1)) \tag{51}$$

In the next section, we explain how to build the two functions θ and ε, and then the strong component-graph $\mathfrak{G}_s$.

11 Strong component-graph computation

11.1 Connected component computation

A node $K = (X, v)$ is defined by its support X and its value v. Its support X contains one or many leaf-points x. In particular, X is subdivided by the influence zones of these leaf-points, and this subdivision forms a partition

$$X = \bigsqcup_{x \in \Lambda \cap X} \rho(x) \cap \lambda_v(I) \tag{52}$$

In other words, X can be retrieved from the set of the leaf-points it contains.

By definition, all the leaf-points in the support of a node are connected within the thresholded influence zone graph $\mathfrak{Z}^v = (\Lambda^v, \frown_\Lambda^v)$ obtained at value v, defined by

$$\Lambda^v = \{x \in \Lambda \mid I(x) \geqslant v\} \tag{53}$$

$$\frown_\Lambda^v = \{(x, y) \mid v \in \nu((x, y))\} \tag{54}$$

In particular, building the nodes $K = (X, v)$ at a given value $v \in V$ consists of building the connected components of $\mathfrak{Z}^v = (\Lambda^v, \frown_\Lambda^v)$.

The connectedness relation $\leftrightarrow_\Lambda^v$ is the reflexive-transitive closure of the adjacency $\frown_\Lambda^v$. It is convenient to model adjacency and connectedness relations in a Boolean matrix form. Indeed, this will allow us to simply express the connected component computation (namely, a transitive closure problem) as a matrix product procedure.

Let us consider the Boolean square, symmetric matrix A_v of dimension $|\Lambda| \times |\Lambda|$, defined as $A_v = [a_{i,j}^v]_{1 \leq i,j \leq |\Lambda|}$ with

$$a_{i,j}^v = \begin{cases} 1 & \text{if } x_i \frown_\Lambda^v x_j \\ 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{55}$$

where $\Lambda = \{x_i\}_{i=1}^{|\Lambda|}$. This matrix models the reflexive closure of the adjacency relation $\frown_\Lambda^v$.

The induced connectedness relation —that defines the connected components of leaf-points, and thus the supports of the nodes at value v— is modelled by the Boolean square, symmetric matrix C_v of dimension $|\Lambda| \times |\Lambda|$, defined as $C_v = [c_{i,j}^v]_{1 \leq i,j \leq |\Lambda|}$ with

$$c_{i,j}^v = \begin{cases} 1 & \text{if } x_i \leftrightarrow_\Lambda^v x_j \\ 0 & \text{otherwise} \end{cases} \tag{56}$$

and we compute C_v as

$$C_v = \lim_n A_v^n \tag{57}$$

The time complexity for computing C_v is $\mathcal{O}(|\Lambda|^3)$, that is the standard time complexity for the transitive closure of a graph.

Remark 13 *The matrix A_v is sparse. In particular, it has empty rows and columns for all the labels k such that $a_{k,k} = 0$. These labels k correspond to leaf-points x_k such that $I(x_k) \ngeqslant v$. This allows us to reduce the $|\Lambda|$ term to $|\{x \in \Lambda \mid I(x) \geqslant v\}|$ in $\mathcal{O}(|\Lambda|^3)$.*

Remark 14 *The matrix A_v is Boolean. As a consequence, the matrix product involves Boolean operators $\wedge$ and $\vee$ instead of $.$ and $+$. Consequently, the result of the calculus of an element is 1 as soon as one of the operands of the $\vee$ combination between the $|\Lambda|$ binary terms in $\wedge$ is 1. This also allows us to reduce the time cost of the overall computation by avoiding useless elementary operations.*

The time complexity of the C_v computation can be optimized by considering the structure of the $\trianglelefteq_s$ relation, that is directly linked to that of $\leqslant$.

Property 13 *Let $v, w \in V$. Let us assume that $v \leqslant w$. Let Y be a connected component of $\mathfrak{Z}^w$. Then, there exists a connected component X of $\mathfrak{Z}^v$ such that $Y \subseteq X$.*

The next property is an immediate corollary.

Property 14 *Let $v, w \in V$. Let us assume that $v \prec w$. Let $K' = (Y, w) \in \Theta$ be a node of $\mathfrak{G}_s$. Then, there exists a node $K = (X, v) \in \Theta$ such that $K' \blacktriangleleft_s K$.*

As a consequence, for any $v \in V$, the computation of C_v (that is done from A_v, in theory) can be carried out recursively by determining the transitive closure of the matrix $B_v = [b^v_{i,j}]_{1 \leq i,j \leq |\Lambda|}$ of dimension $|\Lambda| \times |\Lambda|$ defined as follows

$$b^v_{i,j} = \Big(\bigvee_{v \prec w} c^w_{i,j} \Big) \vee \big(v \in \nu^{\nabla}((x_i, x_j)) \big) \vee (i = j \wedge I(x_i) = v) \quad (58)$$

The computation of B_v simply consists of combining the connected component matrices at the directly greater values, and then adding two kinds of information: on the one hand, the adjacency links at value v between pairs of leaf-points (these links justify a posteriori the definition of the influence zone graph); on the other hand the reflexive links associated to the leaf-points that correspond to leaves at value v.

Then, we compute C_v as

$$C_v = \lim_n B^n_v \quad (59)$$

This computation converges more rapidly than the transitive closure of A_v, since a part of the calculus was already carried out in the C_w matrices ($v \prec w$).

Remark 15 *In B_v, the lines and columns k such that $x_{k,k}$ is a leaf-point that satisfies $I(x_{k,k}) = v$ necessarily contain 0 values everywhere but in (k, k). As a consequence, these rows and columns can be omitted in the computation of B_v, and updated only after the transitive closure calculus. This allows to reduce the overall time complexity for building C_v.*

11.2 Building the nodes of the component-graph

Let $v \in V$, and let us suppose that the matrix C_v has been computed. Each non-empty row of C_v corresponds to a given node of Θ at value v. More precisely, for the row $k \in [\![1, |\Lambda|]\!]$, this node is $K = (X, v)$ where

$$X = \bigcup_{\substack{1 \leq i \leq |\Lambda| \\ c^v_{(k,i)} \neq 0}} \rho(x_i) \cap \lambda_v(I) \quad (60)$$

Practically, any node K can be modelled by its canonical label ℓ_K. Each node is then identified by the ℓ_K-th row. Any such row is empty in the lower triangular matrix, that is

$$\ell_K = \min\{i \in [\![1, |\Lambda|]\!] \mid c^v_{(\ell_K,i)} = 1\} \quad (61)$$

Determining the nodes of Θ at value v is then simply done by scanning the lines of C_v. More precisely, this research can be restricted to the rows that already correspond to canonical labels of nodes at values $w \in V$ with $w \prec v$ (plus, of course, the labels ℓ_x of leaf-points x that satisfy $I(x) = v$, which correspond to leaves $(\{x\}, I(x))$).

Following the definition of the function θ (Equation (48)), we finally define the set of nodes of Θ at value v as

$$\theta(v) = \Big\{ k \in [\![1, |\Lambda|]\!] \mid (1 - c^v_{(k,k)}) \vee \bigvee_{i=1}^{k-1} c^v_{(k,i)} = 0 \Big\} \quad (62)$$

Remark 16 *The identification of the nodes of $\ddot{\Theta}$ can be made by scanning, for each node $K = (X, v)$ of label ℓ_K, if there exists a leaf-point $x \in X$ (i.e., such that $c^v_{(\ell_K,\ell_x)} = 1$) with a point $y \in \rho(x)$ such that $I(y) = v$.*

11.3 Building the edges of the component-graph

Let $K' = (Y, w)$ be a node of Θ at value w, and let us suppose that we have computed the set $\theta(v)$ of the nodes at value v with $v \prec w$ and the associated matrix C_v.

There is exactly one node $K = (X, v)$ such that $K' \blacktriangleleft_s K$. By construction of C_v, the canonical label ℓ_K of this node is defined as

$$\ell_K = \min\{i \in [\![1, \Lambda]\!] \mid c^v_{(\ell_{K'},i)} = 1\} \quad (63)$$

Then, following the definition of the function ε (Equation (49)), we define the set of edges of $\blacktriangleleft_s$ between the values $v \prec w$ as

$$\varepsilon((v, w)) = \{(\ell_x, \ell_y) \mid \ell_y \in \theta(w), \ell_x = \min_i \{c^v_{(\ell_y,i)} = 1\}\} \quad (64)$$

An example of whole computation of the connected components of the thresholded influence zone graphs $\mathfrak{Z}^v$ associated to the influence zone graph $\mathfrak{Z}$ of Figure 13 is proposed in Appendix A. In the same appendix, we also derive from these results the θ and ε functions that allow us to model the strong component-graph $\mathfrak{G}_s$ of the image I of Figures 3 and 7.

12 Data structure

Finally, the strong component-graph $\mathfrak{G}_s$ can then be modelled and stored as the enriched Hasse diagram $(V, \prec, \theta, \varepsilon)$. The space costs of θ and ε are the same as that of Θ and $\blacktriangleleft_s$, respectively, since each node is modelled by its canonical label, while each edge is modelled

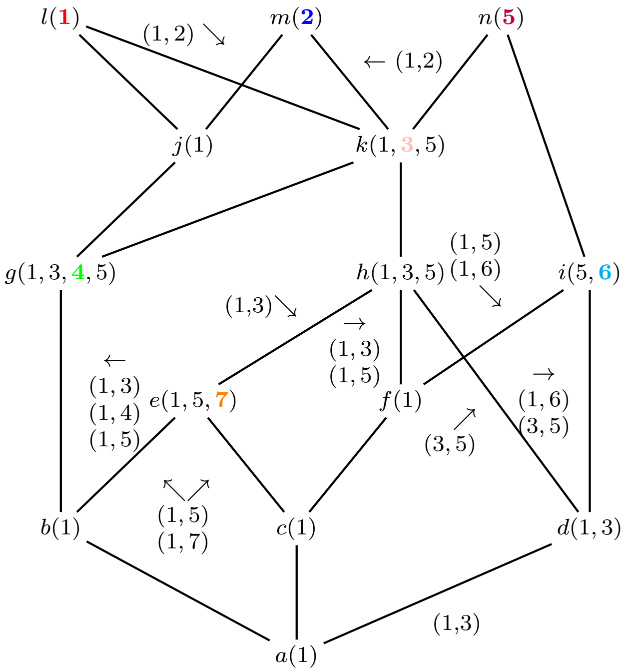

Fig. 14 The strong component-graph $\mathfrak{G}_s$ of the image I of Figures 3 and 7. This component-graph is the same as in Figure 5. However, it is expressed here as the Hasse diagram of $(V, \leqslant)$ enriched with the functions θ and ε. For each value $v \in V$, the labels of the set $\theta(v)$ are provided into brackets. For each couple of adjacent values $v \prec w$, the elements of $\varepsilon((v, w))$ are provided as a valuation of the edge between v and w. Note that when a couple (x, x) belongs to $\varepsilon((v, w))$, it is useless to represent it, as we necessarily have $x \in \theta(v)$ and $x \in \theta(w)$. The labels corresponding to leaves are provided in bold and colour fonts.

by a couple of such labels. The overall cost of the induced data structure is then $\mathcal{O}(|V| + |{\prec}| + |\Theta| + |{\blacktriangleleft_s}|)$, which is generally equal to $\mathcal{O}(|{\blacktriangleleft_s}|)$.

Figure 14 illustrates the strong component-graph $\mathfrak{G}_s$ of the image I of Figures 3 and 7, in this paradigm of enriched Hasse diagram.

Such data structure describes the nodes from a symbolic point of view, i.e. by their canonical labels. This allows one to carry out a structural analysis of the component-graph, for instance by observing the modifications of topology between the nodes. (By analogy with the component-tree, this would correspond to observing the bifurcations between branches, and the endings of the branches.)

However, since a node $K = (X, v)$ is stored as $K = (\ell_K, v)$, we do not have a direct access to X. The information X can be retrieved by various ways:

- by first determining all the leaf-points $x \in X$, and then reconstruction X by a seeded-region growing in $\lambda_v(I)$; or
- by directly collecting the points $x \in X$ that need then to be stored in $\mathfrak{G}_s$.

The second option requires to store, at each node K not only its canonical label ℓ_K, but also all the points $x \in X$ such that $I(x) = v$. This induces an extra space cost of $\mathcal{O}(\Omega)$. In addition the time cost for building one X is then $\mathcal{O}(|K^{\downarrow}|)$, since all the nodes below K have to be scanned for collecting the points x at each value greater than v.

The first option requires either to store all the labels of the leaf-points $x \in X$, or to retrieve them by scanning the nodes below K. In the first case, the overall extra space cost is $\mathcal{O}(\sum_{x \in A} L_x^{\uparrow})$, and the time cost for building one X is $\mathcal{O}(|X|)$. In the second case, there is no extra space cost, but the time cost for building X is $\mathcal{O}(|X| + |K^{\downarrow}|)$.

Remark 17 *The computation of the supports X of the nodes may also be carried out on the flight during the processing of the component-graph, in the cases where such processing requires to scan the data structure in a bottom-up fashion, i.e. from the leaves up to the root.*

13 Conclusion

In this article, we explained how to build the component-graph of an image taking its values in any (totally or partially) ordered set. In particular, we developed strategies for reducing as much as possible the space cost of the processed data, by considering first a flat zone model of the image, and second an influence zone model based on the local maxima of the image. This led to the symbolic representation of the image as an influence zone graph. Then, we observed that the strong component-graphs have structural regularity properties, compared to the Hasse diagram of the considered ordered value set. This regularity allowed us to develop a recursive strategy for building the connected components at each value. Indeed, this step of connected component construction is the most costly (with a polynomial complexity). Then, reducing both the size of the input (thanks to the notion of influence zone graph) and the time cost (thanks to incremental computation of the connected components and efficient calculus on Boolean matrices) is a relevant way to decrease the time complexity of this crucial step. Finally, we also emphasized how a component-graph could be modelled and stored by enriching the Hasse diagram of the associated ordered set of values.

Building component-graphs in a simple and efficient fashion opens the way to their actual involvement in image processing and analysis procedures. In particular, our next works will consist of extending the standard attribute-based antiextensive filtering process [2, 19] initially developed for component-trees, in the case of grey-level images, in order to make it tractable with component-graphs, in the case of multivalued images.

Acknowledgements

The research leading to these results was partly funded by the French *Agence Nationale de la Recherche* (Grant agreement ANR-15-CE23-0009).

A Computation of the connected components

For the 3 maximal values $v = l$, m and n of V, the 3 matrices B_v are defined only by terms $b^v_{i,j} = (i = j \wedge I(x_i) = v)$ (Equation (58)). In other words, they only carry information corresponding to leaves. We have

$$C_n = B^1_n = \begin{pmatrix} 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0 \end{pmatrix} \quad (65)$$

$$C_m = B^1_m = \begin{pmatrix} 0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0 \end{pmatrix} \quad (66)$$

$$C_l = B^1_l = \begin{pmatrix} 1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0 \end{pmatrix} \quad (67)$$

We then have one valued connected component for each of the three values, namely $(5, n)$, $(2, m)$ and $(1, l)$. In other words, we have $\theta(n) = \{5\}$, $\theta(m) = \{2\}$ and $\theta(l) = \{1\}$.

For $v = k$, the matrix B_k is defined as the disjunction of the three matrices C_l, C_m and C_n. This leads to $b^k_{1,1} = b^k_{2,2} = b^k_{5,5} = 1$. Moreover, we have $k \in \nu^\nabla((x_1, x_2))$, that implies $b^j_{1,2} = b^j_{2,1} = 1$. In addition, the leaf-point x_3 satisfies $I(x_3) = k$; then, we set $b^k_{1,1} = 1$. We have

$$C_k = B^1_k = \begin{pmatrix} 1&1&0&0&0&0&0\\ 1&1&0&0&0&0&0\\ 0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0 \end{pmatrix} \quad (68)$$

Then, we have three valued connected components, namely $(1, k)$, $(3, k)$ and $(5, k)$. In other words, we have $\theta(k) = \{1, 3, 5\}$. From the analysis of C_l, C_m and C_n, it also comes $\varepsilon((k, l)) = \{(1, 1)\}$, $\varepsilon((k, m)) = \{(1, 2)\}$ and $\varepsilon((k, n)) = \{(5, 5)\}$.

For $v = j$, the matrix B_j is defined as the disjunction of the two matrices C_l and C_m. Moreover, we have $j \in \nu^\nabla((x_1, x_2))$, that implies $b^j_{1,2} = b^j_{2,1} = 1$. We have

$$C_j = B^1_j = \begin{pmatrix} 1&1&0&0&0&0&0\\ 1&1&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0 \end{pmatrix} \quad (69)$$

Then, we have one valued connected component, namely $(1, j)$. In other words, we have $\theta(j) = \{1\}$. From the analysis of C_l and C_m, it also comes $\varepsilon((j, l)) = \{(1, 1)\}$ and $\varepsilon((j, m)) = \{(1, 2)\}$.

For $v = i$, the matrix B_i is defined from the matrix C_n. Moreover, the leaf-point x_6 satisfies $I(x_6) = i$; then, we set $b^k_{6,6} = 1$. We have

$$C_i = B^1_i = \begin{pmatrix} 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0\\ 0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0 \end{pmatrix} \quad (70)$$

Then, we have two valued connected components, namely $(5, i)$ and $(6, i)$. In other words, we have $\theta(i) = \{5, 6\}$. From the analysis of C_n, it also comes $\varepsilon((i, n)) = \{(5, 5)\}$.

For $v = h$, the matrix B_h is defined from the matrix C_k. We have

$$C_h = B^1_h = \begin{pmatrix} 1&1&0&0&0&0&0\\ 1&1&0&0&0&0&0\\ 0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0 \end{pmatrix} \quad (71)$$

Then, we have 3 valued connected components, namely $(1, h)$, $(3, h)$ and $(5, h)$. In other words, we have $\theta(h) = \{1, 3, 5\}$. From the analysis of C_k, it also comes $\varepsilon((h, k)) = \{(1, 1), (3, 3), (5, 5)\}$.

For $v = g$, the matrix B_g is defined as the disjunction of the two matrices C_j and C_k. Moreover, the leaf-point x_4 satisfies $I(x_4) = g$; then, we set $b^g_{4,4} = 1$. We have

$$C_g = B^1_g = \begin{pmatrix} 1&1&0&0&0&0&0\\ 1&1&0&0&0&0&0\\ 0&0&1&0&0&0&0\\ 0&0&0&1&0&0&0\\ 0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0 \end{pmatrix} \quad (72)$$

Then, we have 4 valued connected components, namely $(1, g)$, $(3, g)$, $(4, g)$ and $(5, g)$. In other words, we have $\theta(g) = \{1, 3, 4, 5\}$. From the analysis of C_j and C_k, it also comes $\varepsilon((g, j)) = \{(1, 1)\}$ and $\varepsilon((g, k)) = \{(1, 1), (3, 3), (5, 5)\}$.

For $v = f$, the matrix B_f is defined as the disjunction of the two matrices C_h and C_i. Moreover, we have

$f \in \nu^{\nabla}((x_1, x_6))$, that implies $b^f_{1,6} = b^f_{6,1} = 1$. The same holds for $\nu^{\nabla}((x_2, x_3))$ and $\nu^{\nabla}((x_3, x_5))$. We have

$$B_f = \begin{pmatrix} 1&1&0&0&0&1&0\\ 1&1&1&0&0&0&0\\ 0&1&1&0&1&0&0\\ 0&0&0&0&0&0&0\\ 0&0&1&0&1&0&0\\ 1&0&0&0&0&1&0\\ 0&0&0&0&0&0&0 \end{pmatrix} \qquad (73)$$

$$B_f^2 = \begin{pmatrix} 1&1&1&0&0&1&0\\ 1&1&1&0&1&1&0\\ 1&1&1&0&1&0&0\\ 0&0&0&0&0&0&0\\ 0&1&1&0&1&0&0\\ 1&1&0&0&0&1&0\\ 0&0&0&0&0&0&0 \end{pmatrix} \qquad (74)$$

$$C_f = B_f^4 = \begin{pmatrix} 1&1&1&0&1&1&0\\ 1&1&1&0&1&1&0\\ 1&1&1&0&1&1&0\\ 0&0&0&0&0&0&0\\ 1&1&1&0&1&1&0\\ 1&1&1&0&1&1&0\\ 0&0&0&0&0&0&0 \end{pmatrix} \qquad (75)$$

Then, we have one valued connected component, namely $(1, f)$. In other words, we have $\theta(f) = \{1\}$. From the analysis of C_h and C_i, it also comes $\varepsilon((f, h)) = \{(1, 1), (1, 3), (1, 5)\}$ and $\varepsilon((f, i)) = \{(1, 5), (1, 6)\}$.

For $v = e$, the matrix B_e is defined from the matrix C_h. Moreover, the leaf-point x_7 satisfies $I(x_7) = e$; then, we set $b^e_{7,7} = 1$. Moreover, we have $e \in \nu^{\nabla}((x_2, x_3))$, that implies $b^e_{2,3} = b^e_{3,2} = 1$. We have

$$B_e = \begin{pmatrix} 1&1&0&0&0&0&0\\ 1&1&1&0&0&0&0\\ 0&1&1&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1 \end{pmatrix} \qquad (76)$$

$$C_e = B_e^2 = \begin{pmatrix} 1&1&1&0&0&0&0\\ 1&1&1&0&0&0&0\\ 1&1&1&0&0&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1 \end{pmatrix} \qquad (77)$$

Then, we have 3 valued connected components, namely $(1, e)$, $(5, e)$ and $(7, e)$. In other words, we have $\theta(e) = \{1, 5, 7\}$. From the analysis of C_h, it also comes $\varepsilon((e, h)) = \{(1, 1), (1, 3), (5, 5)\}$.

For $v = d$, the matrix B_d is defined as the disjunction of the two matrices C_h and C_i. Moreover, we have $d \in \nu^{\nabla}((x_1, x_6))$, that implies $b^d_{1,6} = b^d_{6,1} = 1$, and this is also the case for $\nu^{\nabla}((x_3, x_5))$. We have

$$B_d = \begin{pmatrix} 1&1&0&0&0&1&0\\ 1&1&0&0&0&0&0\\ 0&0&1&0&1&0&0\\ 0&0&0&0&0&0&0\\ 0&0&1&0&1&0&0\\ 1&0&0&0&0&1&0\\ 0&0&0&0&0&0&0 \end{pmatrix} \qquad (78)$$

$$C_d = B_d^2 = \begin{pmatrix} 1&1&0&0&0&1&0\\ 1&1&0&0&0&1&0\\ 0&0&1&0&1&0&0\\ 0&0&0&0&0&0&0\\ 0&0&1&0&1&0&0\\ 1&1&0&0&0&1&0\\ 0&0&0&0&0&0&0 \end{pmatrix} \qquad (79)$$

Then, we have two valued connected components, namely $(1, d)$ and $(3, d)$. In other words, we have $\theta(d) = \{1, 3\}$. From the analysis of C_h and C_i, it also comes $\varepsilon((d, h)) = \{(1, 1), (3, 3), (3, 5)\}$ and $\varepsilon((d, i)) = \{(1, 6), (3, 5)\}$.

For $v = c$, the matrix B_c is defined as the disjunction of the two matrices C_e and C_f. Moreover, we have $c \in \nu^{\nabla}((x_2, x_7))$, $\nu^{\nabla}((x_5, x_6))$ and $\nu^{\nabla}((x_5, x_7))$. This implies $b^c_{5,6} = b^c_{6,5} = 1$ (which was already the case), but also $b^c_{2,7} = b^c_{7,2} = 1$ and $b^c_{5,7} = b^c_{7,5} = 1$. We have

$$B_c = \begin{pmatrix} 1&1&1&0&1&1&0\\ 1&1&1&0&1&1&1\\ 1&1&1&0&1&1&0\\ 0&0&0&0&0&0&0\\ 1&1&1&0&1&1&1\\ 1&1&1&0&1&1&0\\ 0&1&0&0&1&0&1 \end{pmatrix} \qquad (80)$$

$$C_c = B_c^2 = \begin{pmatrix} 1&1&1&0&1&1&1\\ 1&1&1&0&1&1&1\\ 1&1&1&0&1&1&1\\ 0&0&0&0&0&0&0\\ 1&1&1&0&1&1&1\\ 1&1&1&0&1&1&1\\ 1&1&1&0&1&1&1 \end{pmatrix} \qquad (81)$$

Then, we have one valued connected component, namely $(1, c)$. In other words, we have $\theta(c) = \{1\}$. From the analysis of C_e and C_f, it also comes $\varepsilon((c, e)) = \{(1, 1), (1, 5), (1, 7)\}$ and $\varepsilon((c, f)) = \{(1, 1)\}$.

For $v = b$, the matrix B_b is defined as the disjunction of the two matrices C_e and C_g. Moreover, we have $b \in \nu^{\nabla}((x_1, x_4))$, that implies $b^j_{1,4} = b^j_{4,1} = 1$, and this is also the case for $\nu^{\nabla}((x_3, x_4))$, $\nu^{\nabla}((x_3, x_5))$ and $\nu^{\nabla}((x_2, x_7))$. We have

$$B_b = \begin{pmatrix} 1&1&1&1&0&0&0\\ 1&1&1&0&0&0&1\\ 1&1&1&1&1&0&0\\ 1&0&1&1&0&0&0\\ 0&0&1&0&1&0&0\\ 0&0&0&0&0&0&0\\ 0&1&0&0&0&0&1 \end{pmatrix} \qquad (82)$$

$$B_b^2 = \begin{pmatrix} 1&1&1&1&1&0&1\\ 1&1&1&1&1&0&1\\ 1&1&1&1&1&0&1\\ 1&1&1&1&1&0&0\\ 1&1&1&1&1&0&0\\ 0&0&0&0&0&0&0\\ 1&1&1&0&0&0&1 \end{pmatrix} \qquad (83)$$

$$C_b = B_b^4 = \begin{pmatrix} 1&1&1&1&1&0&1\\ 1&1&1&1&1&0&1\\ 1&1&1&1&1&0&1\\ 1&1&1&1&1&0&1\\ 1&1&1&1&1&0&1\\ 0&0&0&0&0&0&0\\ 1&1&1&1&1&0&1 \end{pmatrix} \qquad (84)$$

Then, we have one valued connected component, namely $(1, b)$. In other words, we have $\theta(b) = \{1\}$. From the analysis of C_e and C_g, it also comes $\varepsilon((b,e)) = \{(1,1),(1,5),(1,7)\}$ and $\varepsilon((b,g)) = \{(1,1),(1,3),(1,4),(1,5)\}$.

For $v = a$, we necessarily have

$$C_a = \begin{pmatrix} 1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1 \end{pmatrix} \quad (85)$$

Then, we have one valued connected component, namely $(1,a)$. In other words, we have $\theta(a) = \{1\}$. From the analysis of C_b, C_c and C_d, it also comes $\varepsilon((a,b)) = \{(1,1)\}$, $\varepsilon((a,c)) = \{(1,1)\}$ and $\varepsilon((a,d)) = \{(1,1),(1,3)\}$.

References

1. Najman, L., Talbot, H. (eds.): Mathematical Morphology: From Theory to Applications. ISTE/J. Wiley & Sons (2010)
2. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. IEEE Transactions on Image Processing **7**, 555–570 (1998)
3. Monasse, P., Guichard, F.: Scale-space from a level lines tree. Journal of Visual Communication and Image Representation **11**, 224–236 (2000)
4. Salembier, P., Garrido, L.: Binary partition tree as an efficient representation for image processing, segmentation and information retrieval. IEEE Transactions on Image Processing **9**, 561–576 (2000)
5. Passat, N., Naegel, B.: Component-hypertrees for image segmentation. In: ISMM, International Symposium on Mathematical Morphology, Proceedings, *Lecture Notes in Computer Science*, vol. 6671, pp. 284–295. Springer (2011)
6. Perret, B., Cousty, J., Tankyevych, O., Talbot, H., Passat, N.: Directed connected operators: Asymmetric hierarchies for image filtering and segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(6), 1162–1176 (2015)
7. Naegel, B., Passat, N.: Component-trees and multi-value images: A comparative study. In: International Symposium on Mathematical Morphology (ISMM), *Lecture Notes in Computer Science*, vol. 5720, pp. 261–271. Springer (2009)
8. Passat, N., Naegel, B.: An extension of component-trees to partial orders. In: International Conference on Image Processing (ICIP), pp. 3981–3984 (2009)
9. Passat, N., Naegel, B.: Component-trees and multivalued images: Structural properties. Journal of Mathematical Imaging and Vision **49**, 37–50 (2014)
10. Kurtz, C., Naegel, B., Passat, N.: Connected filtering based on multivalued component-trees. IEEE Transactions on Image Processing **23**, 5152–5164 (2014)
11. Naegel, B., Passat, N.: Toward connected filtering based on component-graphs. In: International Symposium on Mathematical Morphology (ISMM), *Lecture Notes in Computer Science*, vol. 7883, pp. 350–361. Springer (2013)
12. Naegel, B., Passat, N.: Colour image filtering with component-graphs. In: International Conference on Pattern Recognition (ICPR), pp. 1621–1626 (2014)
13. Carlinet, E., Géraud, T.: MToS: A tree of shapes for multivariate images. IEEE Transactions on Image Processing **24**, 5330–5342 (2015)
14. Xu, Y., Géraud, T., Najman, L.: Connected filtering on tree-based shape-spaces. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**, 1126–1140 (2016)
15. Grossiord, É., Naegel, B., Talbot, H., Passat, N., Najman, L.: Shape-based analysis on component-graphs for multivalued image processing. In: International Symposium on Mathematical Morphology (ISMM), *Lecture Notes in Computer Science*, vol. 9082, pp. 446–457. Springer (2015)
16. Grossiord, É., Naegel, B., Talbot, H., Najman, L., Passat, N.: Shape-based analysis on component-graphs for multivalued image processing. HAL Research Report, hal-01695384 (2018)
17. Passat, N., Naegel, B., Kurtz, C.: Implicit component-graph: A discussion. In: International Symposium on Mathematical Morphology (ISMM), *Lecture Notes in Computer Science*, vol. 10225, pp. 235–248. Springer (2017)
18. Salembier, P., Serra, J.: Flat zones filtering, connected operators, and filters by reconstruction. IEEE Transactions on Image Processing **4**, 1153–1160 (1995)
19. Jones, R.: Connected filtering and segmentation using component trees. Computer Vision and Image Understanding **75**(3), 215–228 (1999)