



HAL
open science

Building the topological tree of shapes from the tree of shapes

Julien Mendes Forte, Nicolas Passat, Yukiko Kenmochi

► **To cite this version:**

Julien Mendes Forte, Nicolas Passat, Yukiko Kenmochi. Building the topological tree of shapes from the tree of shapes. International Conference on Discrete Geometry and Mathematical Morphology (DGMM), 2024, Florence, Italy. pp.271-285, 10.1007/978-3-031-57793-2_21 . hal-04363684

HAL Id: hal-04363684

<https://hal.science/hal-04363684v1>

Submitted on 19 Jan 2024

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.

Building the Topological Tree of Shapes from the Tree of Shapes[★]

Julien Mendes Forte¹, Nicolas Passat²[0000-0002-0320-4581], and
Yukiko Kenmochi¹[0000-0001-9648-326X]

¹ Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, F-14050 Caen, France

² Université de Reims Champagne-Ardenne, CReSTIC, 51100 Reims, France

Abstract. The topological tree of shapes was recently introduced as a new hierarchical structure within the family of morphological trees. Morphological trees are efficient models for image processing and analysis. For such applications, it is of paramount importance that these structures be built and manipulated with optimal complexity. In this article, we focus on the construction of the topological tree of shapes. We propose an algorithm for building the topological tree of shapes from the tree of shapes. In particular, a cornerstone of this algorithm is the construction of the complete tree of shapes, another recently introduced tree unifying both the tree of shapes and the topological tree of shapes. We also discuss the cost of the computation of these structures.

1 Introduction

Graphs are popular mathematical structures for the representation of objects and their relationships. In particular, they are widely used in mathematical morphology, especially in the field of connected operators [26]. In this context, they offer an efficient means for organizing information hierarchically, for modelling and manipulation purpose. The induced structures are often trees which encode hierarchies of partitions.

In mathematical morphology these trees often model the structure of images, with regard to their spatial and spectral information, via the topological organization of the connected components related to their flat zones. In the family of partial partitions, the main two subfamilies of hierarchical structures are component-trees and trees of shapes. The component-tree [25] models the inclusion relation between the connected components derived from the threshold sets of an image. The tree of shapes [16] models the nested inclusion between (the boundaries of) these connected components.

At the confluence of these two subfamilies, the topological tree of shapes was recently introduced [18]. It was initially designed for modelling the two orders—inclusion and nesting—associated to component-trees and trees of shapes. In particular, the topological tree of shapes is structurally derived from the min- and max-trees of an image (two dual component-trees) and the adjacency-tree (a binary variant of the tree of shapes) [24] of each threshold set.

[★] This work was supported by Région Normandie (thesis grant RIN), Partenariats Hubert Curien (grant Sakura 49674RK) and ANR (Grants ANR-22-CE45-0034 and ANR-23-CE45-0015).

The topological tree of shapes and the tree of shapes³ derive (by a decreasing homeomorphism) from the complete tree of shapes, which is also introduced in [18]. This led to a construction scheme of the topological tree of shapes from the tree of shapes, via the complete tree of shapes. A first non-optimal algorithm was proposed in [19].

In this article, we propose an algorithm for building the topological tree of shapes that improves the one of [19]. It aims to minimize the space cost of intermediate structures by avoiding the storage of redundant information. Doing so, it also reduces the time cost for their construction and handling, leading to an overall cost optimization.

This article is organized as follows. Sect. 2 recalls previous works on component-trees and trees of shapes. Sect. 3 provides useful definitions. Sect. 4 describes the algorithmic scheme for building the topological tree of shapes from the tree of shapes. Sect. 5 provides a complexity analysis of this algorithm. Sect. 6 concludes this article.

2 Related works

The trees developed in mathematical morphology can be divided into those that model total partitions (e.g. binary partition trees, watershed trees) or partial partitions. We focus on the second, which mainly include the component tree and the tree of shapes.

The component tree [2, 25] is based on the inclusion of the threshold sets of an image. Initially designed for grey-level images, it also led to variants dedicated to images on partially ordered values (e.g. [14]), and alternative connectivity paradigms (e.g. [22]). It was involved in various applications, mostly for filtering and segmentation based on attribute selection [12] or optimal cut computation [11]. More recently, it was also investigated as a way to embed topological information in deep-learning frameworks [21]. Beyond its potential applications, many efforts were geared towards developing efficient construction algorithms for component trees [4]. More recently, parallel algorithms were proposed, including distributed paradigms [10, 8] and GPU-based approaches [1].

The tree of shapes [16] is based on the nested relation between the level-lines of an image. It is often described as a self-dual version of the component-tree, since these level-lines are defined by the hole-closing of the connected components of the min- and max-trees. The tree of shapes also led to variants dedicated to multivalued images [5], or compact versions focusing on the topological structure of the level-lines [27]. The adjacency tree [24], introduced as a topological descriptor for binary images, is also a variant of tree of shapes. The tree of shapes was involved in various image processing and analysis applications, mainly including segmentation. It is for instance related to the minimum barrier distance in images, and allows in particular to estimate it with various potential applications [17]. Many strategies were proposed for efficiently building the tree of shapes, e.g. based on a union-find structure [9] or via root-to-leave strategies [15]. Some parallel approaches were also investigated [7].

³ The term “shape” as used in this paper originates from the definition of the tree of shapes in [16]. A shape is hence represented by its set of boundaries, with one being its outer boundary and the others (possibly none) being its inner boundaries. In the case of the tree of shapes, only the outer boundary is used to represent a shape. In the case of the complete tree of shapes and the topological tree of shapes, all the (outer and inner) boundaries are considered.

Within the literature dedicated to the construction of the component-tree and the tree of shapes, some algorithms rely on the first to build the second [6, 3] or vice versa [28]. This is motivated by the strong links that exist between them. In [19], we introduced the unifying notion of a complete tree of shapes, that provides a continuum between both structures. The complete tree of shapes contains the nodes of the min- and max-trees, and allows to derive the tree of shapes by a decreasing (reversible) homeomorphism. Since the topological tree of shapes can also be derived from the complete tree of shapes by a decreasing (yet non-reversible) homeomorphism, we propose to build the topological tree of shapes from the tree of shapes via the complete tree of shapes.

3 Background notions

We consider images defined on a discrete support \mathbb{U} where the Jordan-Brouwer property holds. In practice, we assume $\mathbb{U} = \mathbb{Z}^d$ ($d \geq 2$), endowed with the digital topology framework [13]. We also consider that images take their values in a finite, totally ordered set (\mathbb{V}, \leq) . Without loss of generality, we assume that $\mathbb{V} = \llbracket \perp, \top \rrbracket \subset \mathbb{Z}$. An image is defined as a function $\mathcal{F} : \mathbb{U} \rightarrow \mathbb{V}$. We assume that the number of points $\mathbf{x} \in \mathbb{U}$ such that $\mathcal{F}(\mathbf{x}) > \perp$ is finite. This number n is considered as the size of the image.

Let $v \in \mathbb{V}$. The upper- and lower-threshold sets of \mathcal{F} (see Fig. 1(a)) at value v are the subsets of \mathbb{U} defined as

$$\begin{aligned} \Lambda_v^\circ(\mathcal{F}) &= \{\mathbf{x} \in \mathbb{U} \mid v \leq \mathcal{F}(\mathbf{x})\} \\ \Lambda_v^\bullet(\mathcal{F}) &= \{\mathbf{x} \in \mathbb{U} \mid v > \mathcal{F}(\mathbf{x})\} \end{aligned} \quad (1)$$

Let $X \subseteq \mathbb{U}$. We note $\Pi[X] \subseteq 2^{\mathbb{U}}$ the set of the connected components of X . For all $v \in \mathbb{V}$, we define the following sets

$$\begin{aligned} \Theta_v^\circ &= \Pi[\Lambda_v^\circ(\mathcal{F})] & \text{and} & & \Theta^\circ &= \bigcup_{v \in \mathbb{V}} \Theta_v^\circ \\ \Theta_v^\bullet &= \Pi[\Lambda_v^\bullet(\mathcal{F})] & & & \Theta^\bullet &= \bigcup_{v \in \mathbb{V}} \Theta_v^\bullet \\ \Theta_v &= \Theta_v^\circ \cup \Theta_v^\bullet & & & \Theta &= \Theta^\circ \cup \Theta^\bullet = \bigcup_{v \in \mathbb{V}} \Theta_v \end{aligned} \quad (2)$$

3.1 Trees

We recall the definition of classic and more recently introduced trees. We consider the partial order relation \subseteq on Θ° (resp. Θ^\bullet) and we note \triangleleft° (resp. \triangleleft^\bullet) the reflexive-transitive reduction of \subseteq on Θ° (resp. Θ^\bullet). (The reflexive-transitive reduction of a binary relation removes the transitive redundancies and ensures that elements are not related with themselves.) We define a tree by a set of elements linked by arcs. Here, these arcs are represented via a relation notation (and not via a set notation).

Definition 1 (Component tree(s) [25]) *The max-tree (resp. min-tree) of \mathcal{F} is the tree $\mathfrak{T}_{\Theta^\circ} = (\Theta^\circ, \triangleleft^\circ)$ (resp. $\mathfrak{T}_{\Theta^\bullet} = (\Theta^\bullet, \triangleleft^\bullet)$). Both trees are also called component-trees.*

Let $X \subset \mathbb{U}$ be a (connected) set. We note $\tau(X) = X^\tau \supseteq X$ the set obtained by closing the holes of X . We note $\Theta^\tau = \{X^\tau \mid X \in \Theta\}$. We consider the partial order relation \subseteq on Θ^τ and we note \triangleleft^τ the reflexive-transitive reduction of \subseteq on Θ^τ .

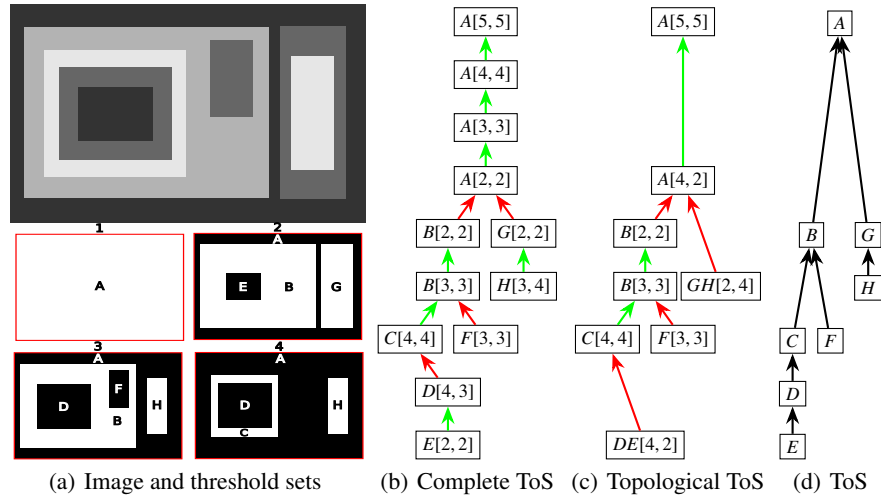


Fig. 1. (a) An image \mathcal{F} (top) and its upper (white) and lower (black) threshold sets (bottom). (b) The complete tree of shapes of \mathcal{F} represents the inclusion (green arcs) or nesting (red arcs) of the connected components of the threshold sets of \mathcal{F} . (c) The topological tree of shapes of \mathcal{F} “compresses” the nodes of the complete tree of shapes with regard to their topological relationship (strong deletability). (d) The tree of shapes of \mathcal{F} represents the nesting of the outer boundaries of the threshold sets of \mathcal{F} . The topological tree of shapes and the tree of shapes are obtained by a decreasing homeomorphism from the complete tree of shapes.

Definition 2 (Tree of shapes [16]) *The tree of shapes of \mathcal{F} is the tree $\mathfrak{T}_{\Theta^\tau} = (\Theta^\tau, \triangleleft^\tau)$. (See Fig. 1(d).)*

Let us suppose that $\mathbb{V} = \{\perp, \top\}$ with $\perp \neq \top$. Then $\mathcal{F} : \mathbb{U} \rightarrow \mathbb{V}$ is a binary image equivalent to the binary set $\Lambda_\top^\circ(\mathcal{F}) \subset \mathbb{U}$. Conversely, any (finite) set $X \subset \mathbb{U}$ is equivalent to a binary function $\mathbf{1}_X : \mathbb{U} \rightarrow \{\perp, \top\}$ defined such that $\mathbf{1}_X(\mathbf{x}) = \top \Leftrightarrow \mathbf{x} \in X$.

Definition 3 (Adjacency tree [24]) *The adjacency tree of a set $X \subset \mathbb{U}$ is the tree of shapes of the binary image $\mathbf{1}_X$.*

Remark 4 *Let $\mathcal{F} : \mathbb{U} \rightarrow \mathbb{V}$, and $v \in \mathbb{V}$. Each upper-threshold set $\Lambda_v^\circ(\mathcal{F})$ of \mathcal{F} is composed by a set of connected components $\Theta_v^\circ = \Pi[\Lambda_v^\circ(\mathcal{F})]$. The adjacency tree of $\Lambda_v^\circ(\mathcal{F})$ is the tree $\mathfrak{T}_{\Theta_v} = (\Theta_v, \triangleleft^v)$ which is the tree of shapes of $\mathbf{1}_{\Lambda_v^\circ(\mathcal{F})}$. This allows us to link the component trees $\mathfrak{T}_{\Theta^\circ}$ and $\mathfrak{T}_{\Theta^\bullet}$, as $(\Theta_v, \triangleleft^v)$ models nested relations between elements of Θ_v° and Θ_v^\bullet for any $v \in \mathbb{V}$.*

Let \sqsubseteq be the partial order relation defined on Θ by $X \sqsubseteq Y \Leftrightarrow (X \subseteq Y) \vee (X^\tau \subseteq Y^\tau)$. We note \triangleleft the reflexive-transitive reduction of \sqsubseteq on Θ .

Definition 5 (Complete tree of shapes [19]) *The complete tree of shapes of \mathcal{F} is the tree $\mathfrak{T}_\Theta = (\Theta, \triangleleft)$. (See Fig. 1(b).)*

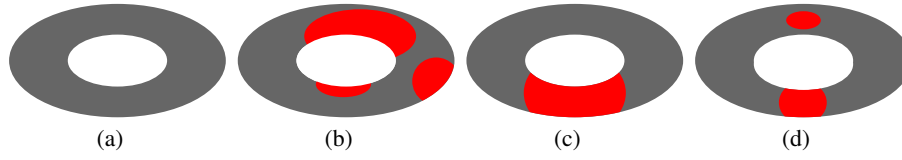


Fig. 2. (a) A set $\Lambda \in \mathbb{U}$. (b–d) The red part of Λ depicts $D \subseteq \Lambda$. (b) The set D is strongly deletable since there is a bijection between the connected components of the set and its complement before and after the removal of D . (c) The number of connected component(s) of the background is different before and after removal. Then the set D is not strongly deletable. (d) The number of connected components is the same for the object and the background before and after removal. However, there is no bijection. Indeed, the hole of the initial object has been merged with the infinite background while a new hole has been created. The set D is not strongly deletable.

Let $D \subset \Lambda \subseteq \mathbb{U}$. Let $\iota : \Pi[\Lambda \setminus D] \rightarrow \Pi[\Lambda]$ and $\bar{\iota} : \Pi[\bar{\Lambda}] \rightarrow \Pi[\bar{\Lambda} \setminus D]$ be the two functions defined by $X \subseteq \iota(X)$ and $Y \subseteq \bar{\iota}(Y)$. We say that D is a strongly deletable set (of Λ) if ι and $\bar{\iota}$ are bijective [23] (see Fig. 2). Let $X, Y \in \Theta$ such that $X \triangleleft_{\Theta} Y$. If X is unique for this property and $Y \setminus X$ is a strongly deletable set of Y , then we note $Y \searrow X$. We note \sim the equivalence relation on Θ derived from \searrow . We note $H = \Theta / \sim$. Let \sqsubseteq_H be the order relation on H defined by $X \sqsubseteq_H Y \Leftrightarrow \bigwedge^{\sqsubseteq} X \sqsubseteq \bigwedge^{\sqsubseteq} Y$. We note \triangleleft^H the reflexive-transitive reduction of \sqsubseteq_H on H .

Definition 6 (Topological tree of shapes [19]) *The topological tree of shapes of \mathcal{F} is the tree $\mathfrak{T}_H = (H, \triangleleft^H)$. (See Fig. 1(c).)*

Property 7 ([19]) *There exists a decreasing homeomorphism from the complete tree of shapes to the tree of shapes (resp. to the topological tree of shapes).*

3.2 Composition of tree nodes

The trees defined above allow to model images. This requires in particular to establish a correspondence between the nodes of these trees and their corresponding regions within the image. These regions are named proper parts.

Definition 8 (Proper part in the component tree(s)) *Let $X \in \Theta^{\circ}$ (the same definition holds for Θ^{\bullet}). The proper part of X in the component-tree $\mathfrak{T}_{\Theta^{\circ}} = (\Theta^{\circ}, \triangleleft^{\circ})$ is defined by $\rho(X, \Theta^{\circ}) = X \setminus \bigcup_{Y \triangleleft^{\circ} X} Y$.*

Definition 9 (Proper part in the tree of shapes) *Let $X \in \Theta^{\tau}$. The proper part of X in the tree of shapes $\mathfrak{T}_{\Theta^{\tau}} = (\Theta^{\tau}, \triangleleft^{\tau})$ is defined by $\rho(X, \Theta^{\tau}) = X \setminus \bigcup_{Y \triangleleft^{\tau} X} Y$.*

Definition 10 (Proper part in the complete tree of shapes) *Let $X \in \Theta$. The proper part of X in the complete tree of shapes $\mathfrak{T}_{\Theta} = (\Theta, \triangleleft)$ is defined by $\rho(X, \Theta) = X \setminus \bigcup_{Y \triangleleft X} Y^{\tau}$.*

Remark 11 *Each point $\mathbf{x} \in \mathbb{U}$ is contained in exactly one $\rho(X, \Theta^{\circ})$ (resp. $\rho(X, \Theta^{\bullet})$, $\rho(X, \Theta^{\tau})$, $\rho(X, \Theta)$). More precisely, $\{\rho(X, \Theta^{\circ}) \mid X \in \Theta^{\circ}\}$, $\{\rho(X, \Theta^{\bullet}) \mid X \in \Theta^{\bullet}\}$ and $\{\rho(X, \Theta^{\tau}) \mid X \in \Theta^{\tau}\}$ are partitions of \mathbb{U} . The set $\{\rho(X, \Theta) \mid X \in \Theta\}$ may not be a partition since some proper parts may be empty.*

4 Building the topological tree of shapes

We now describe an algorithmic scheme for building the topological tree of shapes of an image from a precomputed [9] tree of shapes. This scheme relies on four steps: enrichment of the tree of shapes (Sect. 4.1); construction of the graph of shapes from the enriched tree of shapes (Sect. 4.2); construction of the complete tree of shapes from the graph of shapes and the tree of shapes (Sect. 4.3); construction of the topological tree of shapes from the complete tree of shapes and the graph of shapes (Sect. 4.4).

4.1 Enriching the tree of shapes

The cornerstone of the construction of the topological tree of shapes is the construction of the complete tree of shapes. Indeed, the first is obtained from the second via a decreasing homeomorphism. The complete tree of shapes is made of the set $\Theta = \Theta^\circ \cup \Theta^\bullet$ of the nodes of the min- and max-trees, whereas the tree of shapes is made of the (smaller) set Θ^τ where each node is an equivalence class of nodes of Θ° or Θ^\bullet .

Based on this fact, it is required to assign each node of Θ^τ to the class \circ (resp. \bullet) if it is related to nodes of Θ° (resp. Θ^\bullet). In the tree of shapes, it is usual to associate each node $Y \in \Theta^\tau$ to its ‘‘altitude’’ $Alt(Y) \in \mathbb{V}$, which is characterized by

$$Alt(Y) = \begin{cases} \bigvee \{v \in \mathbb{V} \mid X \in \Lambda_v^\circ(\mathcal{F}) \wedge X^\tau = Y\} & \text{if } Y \text{ is in the class } \circ \\ \bigwedge \{v \in \mathbb{V} \mid X \in \Lambda_v^\bullet(\mathcal{F}) \wedge X^\tau = Y\} & \text{if } Y \text{ is in the class } \bullet \end{cases} \quad (3)$$

where \bigvee and \bigwedge are the supremum and infimum respectively. We assume that the tree of shapes $\mathfrak{T}_{\Theta^\tau}$ is natively endowed with this function $Alt : \Theta^\tau \rightarrow \mathbb{V}$. Then, we can classify each node of Θ^τ into the class \circ or \bullet from Alt and the structural information of $\mathfrak{T}_{\Theta^\tau}$ by defining the function $Class : \Theta^\tau \rightarrow \{\circ, \bullet\}$ as

$$Class(Y) = \begin{cases} \circ & \text{if } Y = \mathbb{U} \\ \circ & \text{if } (Y \triangleleft^\tau X) \wedge (Class(X) = \circ) \wedge (Alt(Y) > Alt(X)) \\ \bullet & \text{if } (Y \triangleleft^\tau X) \wedge (Class(X) = \circ) \wedge (Alt(Y) \leq Alt(X)) \\ \bullet & \text{if } (Y \triangleleft^\tau X) \wedge (Class(X) = \bullet) \wedge (Alt(Y) < Alt(X)) \\ \circ & \text{if } (Y \triangleleft^\tau X) \wedge (Class(X) = \bullet) \wedge (Alt(Y) \geq Alt(X)) \end{cases} \quad (4)$$

From this classification, two useful information can be derived for each node $Y \in \Theta^\tau$: (1) the ‘‘origin’’ of each edge incident to Y , and (2) the interval $\mathbb{I}(Y)$ of values associated to Y . Regarding the origin of each edge ($Y \triangleleft^\tau X$), if $Class(X) = Class(Y)$ (resp. $Class(X) \neq Class(Y)$) then this edge derives from the component trees (resp. the adjacency trees) and will be referred to the function φ (resp. ψ). The functions ψ and φ will be defined in the next section.

The interval $\mathbb{I}(Y) = \llbracket \alpha(Y), \omega(Y) \rrbracket$ of each node $Y \in \Theta^\tau$ can be defined by

$$\mathbb{I}(Y) = \begin{cases} \llbracket \perp, \perp \rrbracket & \text{if } Y = \mathbb{U} \\ \llbracket Alt(X) + 1, Alt(Y) \rrbracket & \text{if } (Y \triangleleft^\tau X) \wedge (Class(Y) = Class(X) = \circ) \\ \llbracket Alt(X) - 1, Alt(Y) \rrbracket & \text{if } (Y \triangleleft^\tau X) \wedge (Class(Y) = Class(X) = \bullet) \\ \llbracket Alt(X), Alt(Y) \rrbracket & \text{if } (Y \triangleleft^\tau X) \wedge (Class(Y) \neq Class(X)) \end{cases} \quad (5)$$

Note that we may have $\alpha(Y) \leq \omega(Y)$ or $\alpha(Y) \geq \omega(Y)$, i.e. the intervals are ‘‘oriented’’.

4.2 Building the graph of shapes

Each $Y \in \Theta^\tau$ of the tree of shapes corresponds to an equivalence class $T(Y)$ of nodes of either Θ° or Θ^\bullet . The nodes $X \in T(Y)$ are characterized by $\tau(X) = Y$. If we assume, without loss of correctness, that the nodes of the component trees are defined at each threshold set, $\mathbb{I}(Y)$ and $T(Y)$ are in bijection, since each node $X \in T(Y)$ is a connected component of the threshold set $\Lambda_v^\circ(\mathcal{F})$ (or $\Lambda_v^\bullet(\mathcal{F})$) for a specific value $v \in \mathbb{I}(Y)$. In other words, $\{Y\} \times \mathbb{I}(Y)$ models a subset of nodes of Θ° (or Θ^\bullet). More generally, $\bigcup_{Y \in \Theta^\tau} \{Y\} \times \mathbb{I}(Y)$ models the set Θ . In particular, each node $X \in \Theta_v \subseteq \Theta$ is modeled by the couple (X^τ, v) ($v \in \mathbb{I}(X^\tau)$).

The set Θ^τ endowed with the set of intervals $\mathbb{I}_{\Theta^\tau} = \{\mathbb{I}(Y) \mid Y \in \Theta^\tau\}$ is a compact model of the nodes of the component trees and the adjacency trees (since $|\Theta^\tau| \leq |\Theta|$). Similarly, the edges of the tree of shapes, i.e. the elements of $\triangleleft_{\Theta^\tau}$, represent some edges of the component trees and the adjacency trees. However, this representation is partial. Our purpose is to build the graph of shapes, which enriches the tree of shapes with additional edges that will model all the edges of both the component trees and the adjacency trees. We first define the two functions φ and ψ that model these edges.

Definition 12 *The function $\varphi : \Theta \rightarrow \Theta$ associates each node of the component trees to its parent. It is defined, for any $X \in \Theta^\circ \setminus \{\mathbb{U}\}$ (resp. $\Theta^\bullet \setminus \{\mathbb{U}\}$) by $X \triangleleft^\circ \varphi(X)$ (resp. $X \triangleleft^\bullet \varphi(X)$). The function $\psi : \Theta \rightarrow \Theta$ associates each node of the adjacency trees to its parent. It is defined, for any $X \in \Theta_v \setminus \{\sqrt{\subseteq} \Theta_v\}$ ($v \in \mathbb{V}$) by $X \triangleleft^v \psi(X)$.*

As Θ and $\bigcup_{Y \in \Theta^\tau} \{Y\} \times \mathbb{I}(Y)$ are in bijection, we can rewrite the functions φ and ψ as

$$\left| \begin{array}{l} \varphi : \Theta^\tau \times \mathbb{V} \rightarrow \Theta^\tau \\ (X^\tau, v) \mapsto Y^\tau \text{ s.t. } \begin{cases} X \triangleleft^\circ Y \in \Theta_{v-1}^\circ & \text{if } X \in \Theta_v^\circ \\ X \triangleleft^\bullet Y \in \Theta_{v+1}^\bullet & \text{if } X \in \Theta_v^\bullet \end{cases} \end{array} \right. \quad (6)$$

$$\left| \begin{array}{l} \psi : \Theta^\tau \times \mathbb{V} \rightarrow \Theta^\tau \\ (X^\tau, v) \mapsto Y^\tau \text{ s.t. } X \triangleleft^v Y \in \Theta_v \end{array} \right. \quad (7)$$

The graph of shapes is defined by $(\Theta^\tau, \triangleleft)$, where the set of edges \triangleleft is partitioned into $\{\triangleleft_\varphi, \triangleleft_\psi\}$ such that \triangleleft_φ (resp. \triangleleft_ψ) contains the edges related to φ (resp. ψ).

Let $X^\tau \in \Theta^\tau$. If $X^\tau \neq \mathbb{U}$ there exists exactly one edge (X^τ, Y^τ) in \triangleleft_φ , i.e. exactly one $Y^\tau \in \Theta^\tau$ such that $X^\tau \triangleleft_\varphi Y^\tau$. This edge (X^τ, Y^τ) defines the function φ for $(X^\tau, \alpha(X^\tau))$ by $\varphi(X^\tau, \alpha(X^\tau)) = Y^\tau$, while the function φ for the other (X^τ, v) with $v \in \mathbb{I}(X^\tau) \setminus \{\alpha(X^\tau)\}$ is defined by $\varphi(X^\tau, v) = X^\tau$ and then need not to be modelled by an edge of \triangleleft_φ . Then, φ is fully defined by Θ^τ endowed with \mathbb{I}_{Θ^τ} and by \triangleleft_φ with a space cost $\mathcal{O}(|\Theta^\tau|)$.

Let $X^\tau \in \Theta^\tau$ and $v \in \mathbb{I}(X^\tau)$. If $X^\tau \neq \mathbb{U}$ then there exist(s) $\sigma(X^\tau)$ edge(s) (X^τ, Y^τ) in \triangleleft_ψ , i.e. there exist(s) $\sigma(X^\tau)$ node(s) $Y^\tau \in \Theta^\tau$ such that $X^\tau \triangleleft_\psi Y^\tau$. Each such edge (X^τ, Y^τ) defines the function ψ for (X^τ, v) for a given $v \in \mathbb{I}(X^\tau)$ by $\psi(X^\tau, v) = Y^\tau$. The function ψ may be constant over successive values $v \in \mathbb{I}(X^\tau)$. In practice, the required number $\sigma(X^\tau)$ of edges of \triangleleft_ψ needed to model ψ for $\{X^\tau\} \times \mathbb{I}(X^\tau)$ satisfies $1 \leq \sigma(X^\tau) \leq |\alpha(X^\tau) - \omega(X^\tau)| + 1$. As a counterpart, each edge $\mathcal{E} = (X^\tau, Y^\tau)$ of \triangleleft_ψ has to be endowed with the interval $\mathbb{I}(\mathcal{E}) = \llbracket \alpha(\mathcal{E}), \omega(\mathcal{E}) \rrbracket$ where it is defined. In other words, the function

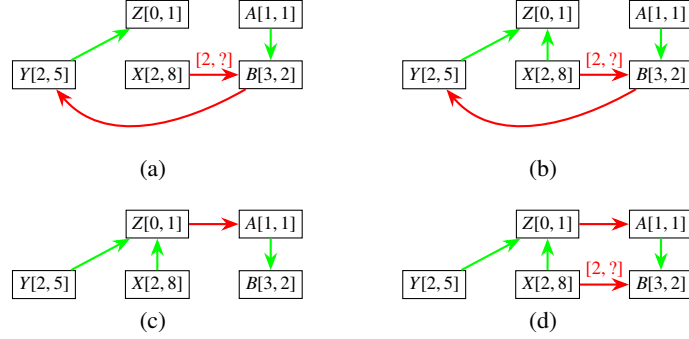


Fig. 3. Partial definition of the edges of $\blacktriangleleft_\varphi$ and \blacktriangleleft_ψ in the graph of shapes from the information φ and ψ in the tree of shapes. The elements A, \dots, Z correspond to nodes of the tree of shapes (Θ^τ). The elements endowed with their interval ($A[1, 1], \dots, Z[0, 1]$) correspond to nodes of the graph of shapes (Θ). The arrows in (a,c) correspond to edges of the tree of shapes; those in (b,d) to edges of the graph of shapes. The green (resp. red) edges relate to φ and $\blacktriangleleft_\varphi$ (resp. ψ and \blacktriangleleft_ψ). We focus on the node X . We recall that the intervals are oriented. As such, depending on the class of a node N , it is possible that $\alpha(N) \leq \omega(N)$ or $\alpha(N) \geq \omega(N)$. In the tree of shapes, X has exactly one outer arc \mathcal{E} which is related to either ψ (a) or φ (c). In case (a), \mathcal{E} is related to \blacktriangleleft_ψ in the graph of shapes (b); the bound $\alpha(\mathcal{E})$ of $\mathbb{I}(\mathcal{E})$ is defined by the bound $\alpha(X)$ of $\mathbb{I}(X)$. In case (c), \mathcal{E} is related to $\blacktriangleleft_\varphi$ in the graph of shapes (d). The transitive relations (Eqs. (8–9)) allow to determine in the graph of shapes the edge φ and $\blacktriangleleft_\varphi$ (b) and the edge ψ and \blacktriangleleft_ψ (d). See Alg. 1, lines 7–14.

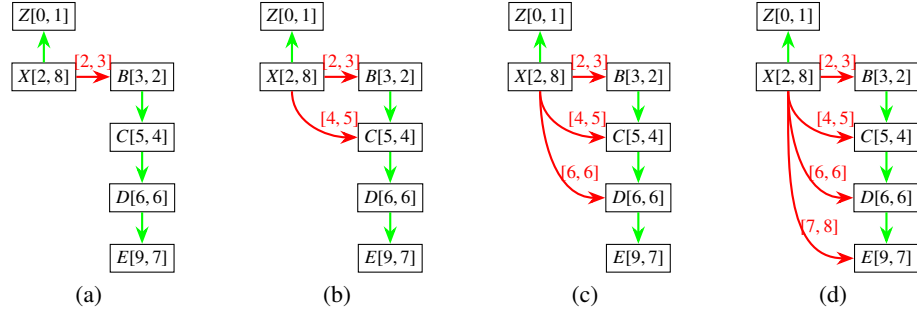


Fig. 4. Complete definition of the edges of \blacktriangleleft_ψ in the graph of shapes. The process starts from a configuration illustrated in Fig. 3(b, d). The successive edges \mathcal{E} of \blacktriangleleft_ψ and their interval $\mathbb{I}(\mathcal{E})$ are obtained from the transitive property of Eq. (9) (a–d). See Alg. 1, 18–25.

ψ is fully defined by \blacktriangleleft_ψ endowed with the set of intervals $\mathbb{I}_{\blacktriangleleft_\psi} = \{\mathbb{I}(\mathcal{E}) \mid \mathcal{E} \in \blacktriangleleft_\psi\}$. Note that $\sum_{X^\tau \in \Theta^\tau} |\sigma(X^\tau)| = \mathcal{O}(|\Theta|)$. The space cost of \blacktriangleleft_ψ is then $\mathcal{O}(|\Theta|)$.

Our purpose is to build the two sets of edges $\blacktriangleleft_\varphi$ and \blacktriangleleft_ψ , in order to obtain the graph of shapes. One important pre-processing step of the algorithm concerns the node $X^\tau = \mathbb{U}$. As initially computed by the tree of shapes, we have $\text{Class}(\mathbb{U}) = \circ$ and $\mathbb{I}(\mathbb{U}) = \llbracket \perp, \perp \rrbracket$. It is however important to note that X^τ actually belongs to both the threshold set $\Lambda_\perp^\circ(\mathcal{F})$ and $\Lambda_\perp^*(\mathcal{F})$, and thus X^τ is part of both classes. Moreover, one node of the

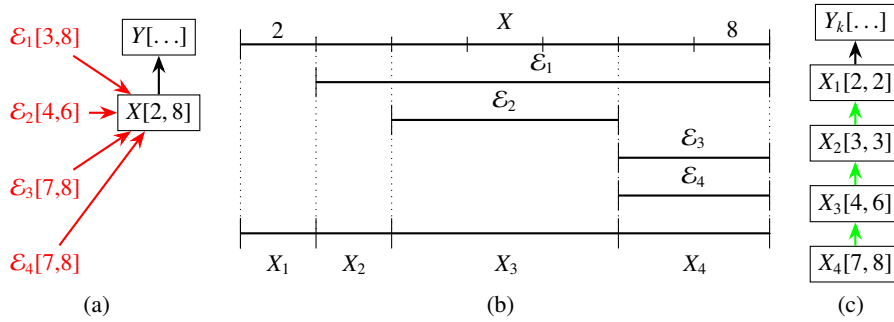


Fig. 5. Definition of the nodes of the complete tree of shapes (c) from the nodes of the graph of shapes (a) and the intervals of the edges of \triangleleft_ψ (b). (a) The node X in the graph of shapes is associated to 4 inner edges \mathcal{E}_i of \triangleleft_ψ with intervals $\mathbb{I}(\mathcal{E}_i)$. (b) These intervals induce a partition of $\mathbb{I}(X)$. This partition allows to define the nodes X_j derived from X in the complete tree of shapes.

equivalence class of X^τ is present in each $\Lambda_v^\bullet(\mathcal{F})$, which is not explicitly encoded by the tree of shapes. Thus, we set $Class(X^\tau) = \bullet$ and $\mathbb{I}(X^\tau) = \llbracket \tau, \perp + 1 \rrbracket$. A part of \triangleleft_φ and \triangleleft_ψ is natively provided by the set of edges $\triangleleft_{\theta^\tau}$ of the tree of shapes. Let $X^\tau \in \Theta^\tau$. If $X^\tau \neq \mathbb{U}$ then there exists exactly one edge (X^τ, Y^τ) in $\triangleleft_{\theta^\tau}$. If $Class(X^\tau) = Class(Y^\tau)$ then this edge defines φ for $(X^\tau, \alpha(X^\tau))$ by $\varphi(X^\tau, \alpha(X^\tau)) = Y^\tau$ and we have $X^\tau \triangleleft_\varphi Y^\tau$. If $Class(X^\tau) \neq Class(Y^\tau)$ then this edge defines ψ for $(X^\tau, \alpha(X^\tau))$ by $\psi(X^\tau, \alpha(X^\tau)) = Y^\tau$ and we have $X^\tau \triangleleft_\psi Y^\tau$. In other words, for each X^τ , either the (unique) edge of \triangleleft_φ is already defined while the $\sigma(X^\tau)$ edges of \triangleleft_ψ remain to be defined, or one edge of \triangleleft_ψ is already defined while the unique edge of \triangleleft_φ and the remaining $\sigma(X^\tau) - 1$ edges \triangleleft_ψ remain to be defined. Fig. 3(a,c) illustrates how these properties allow to initialize the definition of \triangleleft_φ and \triangleleft_ψ in the graph of shapes from the information on φ and ψ carried by the tree of shapes.

These remaining edges to be defined can be computed by a transitive closure procedure that relies on tree properties that link φ and ψ [19]

$$\varphi(X) = [\varphi \circ \psi \circ \psi](X) \quad (8)$$

$$\psi(X) = [\varphi \circ \psi \circ \varphi](X) \quad (9)$$

$$\varphi(X) = [\varphi^{|\mathbb{V}|-2} \circ \psi](X) \quad (10)$$

This transitive closure, illustrated in Fig. 3(b,d) regarding its initialization and Fig. 4 for its iterative part, is performed on the tree of shapes, from its root to its leaves. Note that the third property effectively applies only for the nodes $X^\tau \in \Theta^\tau$ such that $X^\tau \triangleleft_{\theta^\tau} \mathbb{U}$. This is summarized in Alg. 1.

4.3 Building the complete tree of shapes

The tree of shapes can be obtained from the complete tree of shapes by a decreasing homeomorphism [19]. This homeomorphism consists of collapsing the branches of the complete tree of shapes with respect to the equivalence relation on Θ induced by the

Algorithm 1: Construction of the graph of shapes from the tree of shapes.

Input: $\Theta^{\mathcal{F}}, \triangleleft^{\mathcal{F}}, \text{Class}, \mathbb{I}_{\Theta^{\mathcal{F}}}$
Output: $\triangleleft_{\varphi}, \triangleleft_{\psi}, \mathbb{I}_{\triangleleft_{\psi}}$
Notation: “ $A := B$ ” means that A is set with B ; “ $A \leftarrow B$ ” means that B is added to A ;
“ $A \rightarrow B$ ” means that B is set as an element removed from A

```

1  $\mathcal{L} := \{(\cup, \emptyset)\}$  //  $\mathcal{L}$  is a FIFO list
2 while  $\mathcal{L} \neq \emptyset$  do
3    $\mathcal{L} \rightarrow (X, Y)$ 
4   if  $X \neq \cup$  then
5     if  $Y \neq \cup$  then  $Z := \varphi(\psi(Y, \omega(Y)), \omega(Y))$ 
6     else  $Z := Y$ 
7     if  $\text{Class}(X) \neq \text{Class}(Y)$  then
8        $\triangleleft_{\varphi} \leftarrow (X, Z)$ 
9        $\varphi(X, \alpha(X)) := Z$ 
10       $\psi(X, \alpha(X)) := Y$ 
11     else
12        $\triangleleft_{\varphi} \leftarrow (X, Y)$ 
13        $\varphi(X, \alpha(X)) := Y$ 
14        $\psi(X, \alpha(X)) := Z$ 
15      $\widehat{Y} := \psi(X, \alpha(X))$ 
16      $\widehat{\alpha} := \alpha(X)$ 
17     repeat
18        $\mathcal{E} := (X, \widehat{Y})$ 
19        $\triangleleft_{\psi} \leftarrow \mathcal{E}$ 
20        $\widehat{\omega} := \begin{cases} \wedge \{\omega(X), \alpha(\widehat{Y})\} & \text{if } \text{Class}(X) = \circ \\ \vee \{\omega(X), \alpha(\widehat{Y})\} & \text{if } \text{Class}(X) = \bullet \end{cases}$ 
21        $\mathbb{I}_{\triangleleft_{\psi}}(\mathcal{E}) = \llbracket \alpha(\mathcal{E}), \omega(\mathcal{E}) \rrbracket := \llbracket \widehat{\alpha}, \widehat{\omega} \rrbracket$ 
22        $\psi(X, \widehat{\omega}) := \widehat{Y}$ 
23        $\widehat{\alpha} := \begin{cases} \widehat{\omega} + 1 & \text{if } \text{Class}(X) = \circ \\ \widehat{\omega} - 1 & \text{if } \text{Class}(X) = \bullet \end{cases}$ 
24        $\widehat{Y} := \varphi(\widehat{Y}, \alpha(\widehat{Y}))$ 
25     until  $\widehat{\omega} = \omega(X)$ 
26   foreach  $Z \triangleleft_{\Theta^{\mathcal{F}}} X$  do  $\mathcal{L} \leftarrow (Z, X)$ 

```

hole-closing τ (following the same way as for turning the nodes of the component trees into those of the tree of shapes). This homeomorphism is reversible. Each node of the tree of shapes can be duplicated to form a branch of nodes linked by edges of φ . The definition of these branches (by the nodes composing them) is given by the information carried by the relation \triangleleft_{ψ} . This is exemplified by Fig. 5.

Let $X \in \Theta^{\mathcal{F}}$, associated to the interval $\mathbb{I}(X) = \llbracket \alpha(X), \omega(X) \rrbracket$. Let us consider the k edges ($k \in \mathbb{N}$) $\mathcal{E}_i = (Y_i, X)$ ($1 \leq i \leq k$) of \triangleleft_{ψ} that link other nodes Y_i of $\Theta^{\mathcal{F}}$ to X . Each edge \mathcal{E}_i is associated to an interval $\mathbb{I}(\mathcal{E}_i) = \llbracket \alpha(\mathcal{E}_i), \omega(\mathcal{E}_i) \rrbracket$ (Fig. 5(a)). These nodes Y_i define the holes of the nodes of the tree of shapes that derive from X over $\mathbb{I}(X)$. A node is in particular defined by a specific combination of holes Y_i . It follows that the

Algorithm 2: Construction of the complete tree of shapes.

Input: $\Theta^r, \triangleleft^r, \triangleleft_\psi, \mathbb{I}_{\triangleleft_\psi}, \mathbb{I}_{\Theta^r}$
Output: $\Theta, \triangleleft, \mathbb{I}_\Theta = \{\mathbb{I}(X) = \llbracket \alpha(X), \omega(X) \rrbracket \mid X \in \Theta\}$

```

1  $(\Theta, \triangleleft_\Theta) := (\emptyset, \emptyset)$ 
2  $\mathcal{L} := \{(\mathbb{U}, \emptyset)\}$  //  $\mathcal{L}$  is a FIFO list
3 while  $\mathcal{L} \neq \emptyset$  do
4    $\mathcal{L} \rightarrow (X, Y)$ 
5   Build  $\Omega(X) = \{\llbracket \alpha(j), \omega(j) \rrbracket\}_{j=1}^\ell$ 
6   for  $j$  from 1 to  $\ell$  do
7     Build  $X_j$ 
8      $\mathbb{I}(X_j) := \llbracket \alpha(j), \omega(j) \rrbracket$ 
9      $\Theta \leftarrow X_j$ 
10    if  $Y \neq \emptyset$  then  $\triangleleft \leftarrow (X_j, Y)$ 
11     $Y := X_j$ 
12  foreach  $Z \triangleleft^r X$  do  $\mathcal{L} \leftarrow (Z, Y)$ 

```

different nodes of the complete tree of shapes derived from X are in bijection with the partition of intervals induced by the various $\mathbb{I}(E)$, that refines the interval $\mathbb{I}(X)$. We note $\Omega(X) = \{\llbracket \alpha(j), \omega(j) \rrbracket\}_{j=1}^\ell$ (with $1 \leq \ell \leq 2k$) this partition of intervals, with $\alpha(1) = \alpha(X)$, $\omega(\ell) = \omega(X)$, $\alpha(j) \leq \omega(j)$ for all $1 \leq j \leq \ell$ and $\omega(j) + 1 = \alpha(j + 1)$ for all $1 \leq j \leq \ell - 1$ (the same holds with $X \in \Theta^\bullet$ by substituting $\alpha(j) \geq \omega(j)$ to $\alpha(j) \leq \omega(j)$ and $\omega(j) - 1$ to $\omega(j) + 1$), see Fig. 5(b). The partition $\Omega(X)$ can be built in a time $\mathcal{O}(k \log k)$. Once $\Omega(X)$ is built for each node X , the structure of the complete tree of shapes can be obtained from the tree of shapes by substituting a branch composed by a number of nodes equal to the size of $\Omega(X)$ to the node X (Fig. 5(c)). This procedure is summarized in Alg. 2.

The tree generated from Alg. 2 is isomorphic to the complete tree of shapes. However, the nodes of this tree do not contain the information required to link that tree to the image. More precisely, their proper part (Def. 10) is not yet defined. It was proved in [19] that when splitting a node X of the tree of shapes into ℓ nodes X_i of the complete tree of shapes, the “first” node X_1 , characterized by $\alpha(X_1) = \alpha(X)$ inherits the proper part of X , while the other $\ell - 1$ nodes X_i ($2 \leq i \leq \ell$) have an empty proper part. Alg. 2 can be slightly modified (without extra time cost) to deal with this procedure by embedding the required proper part at the creation of each node (line 7).

4.4 Building the topological tree of shapes

The construction of the topological tree of shapes from the complete tree of shapes is a procedure that consists of gathering the nodes of Θ into equivalence classes with respect to the notion of strong deletability. In [19], it was observed that this equivalence between two nodes $X \triangleleft_\varphi Y$ could be assessed by locally observing the structure of their children nodes with respect to \triangleleft_ψ in order to characterize a putative bijection between them. In [19], this checking was carried out in the so-called graph of valued shapes, a non-compact structure equivalent to the current graph of shapes. The characterization established in the graph of valued shapes can be rewritten in the graph of shapes.

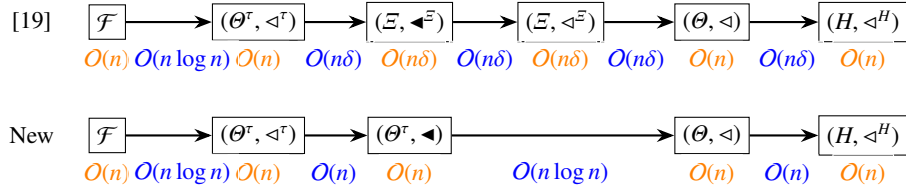


Fig. 6. Synthetic comparison of the two algorithms for building the topological tree of shapes: the former algorithm [19] on the first line and the new algorithm on the second line. The intermediate structures are depicted by the boxes with their space cost in orange. The algorithmic steps for building each structure from the previous are depicted by arrows with their time cost in blue. Both space and time costs are expressed with respect to the size n of the image \mathcal{F} .

Let $X, Y \in \Theta$ be such that $X \triangleleft Y$. If $Y = \psi(X)$ then X and Y cannot be equivalent. Let us assume that $Y = \varphi(X)$. Let X^τ and Y^τ be the nodes of Θ^τ associated to X and Y , respectively. We assume that X^τ is the only node such that $X^\tau \triangleleft_\varphi Y^\tau$ (otherwise, X and Y are not equivalent). Let \mathcal{E}_{X^τ} be the set of edges of \triangleleft_ψ of the form (X_i, X^τ) such that $\alpha((X_i, X^\tau)) = \alpha(X^\tau)$. Let \mathcal{E}_{Y^τ} be the set of edges of \triangleleft_ψ of the form (Y_j, Y^τ) such that $\omega((Y_j, Y^\tau)) = \omega(Y^\tau)$. If these two sets of edges have distinct cardinals, X and Y are not equivalent. Otherwise, X and Y are equivalent iff there is a bijection between the elements X_i and the elements Y_j characterized by $X_i = Y_j$ or $Y_j \triangleleft_\varphi X_i$.

Based on this characterization of the equivalence between the nodes of Θ , we can derive an algorithmic process that involves exactly twice (once for the value $\alpha(\mathcal{E})$ and once for the value $\omega(\mathcal{E})$) each edge of the set \triangleleft_ψ in the graph of shapes. This algorithmic process, which allows to gather the nodes of Θ into equivalence classes leading to the set of nodes H of the topological tree of shapes, then presents a time cost $O(|\Theta|)$.

5 Complexity analysis

The image \mathcal{F} has a space cost n . The construction of the tree of shapes, which is the input of the proposed algorithm, has a time cost $O(n \log n)$ [9]. The space cost of $\mathfrak{T}_{\Theta^\tau}$ is $O(|\Theta^\tau|)$ with $|\Theta^\tau| \leq n$. The enrichment of the tree of shapes (Sect. 4.1), i.e. the construction of $Class(Y)$ and $\mathbb{I}(Y)$ for each $Y \in \Theta^\tau$, and the classification of each edge of \triangleleft^τ into either φ or ψ presents a time cost $O(|\Theta^\tau|)$. The overall space cost of the enriched tree of shapes remains $O(|\Theta^\tau|)$. The construction of the graph of shapes (Sect. 4.2) of size $O(|\Theta|)$ from Alg. 1 has a time cost $O(|\Theta|)$. The construction of the complete tree of shapes (Sect. 4.3) of size $O(|\Theta|)$ from Alg. 2 has a time cost $O(|\Theta| \log |\Theta^\tau|)$. The construction of the topological tree of shapes from the complete tree of shapes and the graph of shapes (Sect. 4.4) has a time cost $O(|\Theta|)$. The overall process of building the topological tree of shapes from the tree of shapes then presents a space cost $O(|\Theta|) = O(n)$ related to the intermediate structures and a time cost $O(|\Theta| \log |\Theta^\tau|) = O(n \log n)$.

This algorithm presents a lower complexity, compared to the former approach in [19] (see Fig. 6). Indeed, we now use the graph of shapes, a compact structure where both redundant nodes and edges are modeled only once and endowed with an interval of definition, while in [19], they were modeled extensively in a structure called the

graph of valued shapes $(\mathcal{E}, \prec^{\mathbb{E}})$ of size $O(n\delta)$ where $\delta = O(|\mathbb{V}|)$ is the average size of the interval of values where the nodes in the component trees of \mathcal{F} exist. The optimization factor in terms of space (resp. time) cost between both algorithms is then δ (resp. $\frac{\delta}{\log n}$).

6 Conclusion

In this article, we continued our study of the topological tree of shapes initiated in [18, 19]. Here, we focused on the design of an efficient algorithm for building the topological tree of shapes from the tree of shapes. We aim to release an implementation of this algorithm in libraries dedicated to mathematical morphology, such as Higras [20]. We presented a theoretical analysis of the cost of this algorithm, that focuses on asymptotic behaviour. Short term perspectives will be to refine this theoretical analysis, and to experimentally assess the cost of its implementation on various datasets of images, to understand how the image properties (size, grey-level range. . .) may influence that cost. As of now, we proposed to build the topological tree of shapes from the tree of shapes. It may be relevant to investigate if/how it may be built ex nihilo from the image.

The topological tree of shapes opens the way to the development of new image processing / analysis operators. On the one hand, the topological tree of shapes can be seen as a rich topological invariant; it may then allow to describe, analyze and/or compare images. On the other hand, as any other tree in the field of morphological hierarchies, it may be used to develop connected operators, e.g. for filtering, segmentation or simplification. Longer term perspectives will consist of investigating such applications.

References

1. Blin, N., Carlinet, E., Lemaitre, F., Lacassagne, L., Géraud, T.: Max-tree computation on GPUs. *IEEE Transactions on Parallel and Distributed Systems* **33**, 3520–3531 (2022)
2. Breen, E.J., Jones, R.: Attribute openings, thinnings, and granulometries. *Computer Vision and Image Understanding* **64**(3), 377–389 (1996)
3. Carlinet, E., Crozet, S., Géraud, T.: The tree of shapes turned into a max-tree: A simple and efficient linear algorithm. In: *ICIP*. pp. 1488–1492 (2018)
4. Carlinet, E., Géraud, T.: A comparative review of component tree computation algorithms. *IEEE Transactions on Image Processing* **23**, 3885–3895 (2014)
5. Carlinet, E., Géraud, T.: MToS: A tree of shapes for multivariate images. *IEEE Transactions on Image Processing* **24**, 5330–5342 (2015)
6. Caselles, V., Meinhardt, E., Monasse, P.: Constructing the tree of shapes of an image by fusion of the trees of connected components of upper and lower level sets. *Positivity* **12**, 55–73 (2008)
7. Crozet, S., Géraud, T.: A first parallel algorithm to compute the morphological tree of shapes of nD images. In: *ICIP*. pp. 2933–2937 (2014)
8. Gazagnes, S., Wilkinson, M.H.F.: Distributed connected component filtering and analysis in 2D and 3D tera-scale data sets. *IEEE Transactions on Image Processing* **30**, 3664–3675 (2021)
9. Géraud, T., Carlinet, E., Crozet, S., Najman, L.: A quasi-linear algorithm to compute the tree of shapes of nD . In: *ISMM*. pp. 98–110 (2013)

10. Götz, M., Cavallaro, G., Géraud, T., Book, M., Riedel, M.: Parallel computation of component trees on distributed memory machines. *IEEE Transactions on Parallel and Distributed Systems* **29**, 2582–2598 (2018)
11. Guigues, L., Cocquerez, J.P., Le Men, H.: Scale-sets image analysis. *International Journal of Computer Vision* **68**, 289–317 (2006)
12. Jones, R.: Connected filtering and segmentation using component trees. *Computer Vision and Image Understanding* **75**, 215–228 (1999)
13. Kong, T.Y., Rosenfeld, A.: Digital topology: Introduction and survey. *Computer Vision, Graphics, and Image Processing* **48(3)**, 357–393 (1989)
14. Kurtz, C., Naegel, B., Passat, N.: Connected filtering based on multivalued component-trees. *IEEE Transactions on Image Processing* **23**, 5152–5164 (2014)
15. Monasse, P.: A root-to-leaf algorithm computing the tree of shapes of an image. In: *RRPR*. pp. 43–54 (2018)
16. Monasse, P., Guichard, F.: Scale-space from a level lines tree. *Journal of Visual Communication and Image Representation* **11(2)**, 224–236 (2000)
17. Ngoc, M.O.V., Boutry, N., Fabrizio, J., Géraud, T.: A minimum barrier distance for multivariate images with applications. *Computer Vision and Image Understanding* **197-198**, 102993 (2020)
18. Passat, N., Kenmochi, Y.: A topological tree of shapes. In: *DGMM*. pp. 221–235 (2022)
19. Passat, N., Mendes Forte, J., Kenmochi, Y.: Morphological hierarchies: A unifying framework with new trees. *Journal of Mathematical Imaging and Vision* **65(5)**, 718–753 (2023)
20. Perret, B., Chierchia, G., Cousty, J., Ferzoli Guimarães, S.J., Kenmochi, Y., Najman, L.: Higr: Hierarchical graph analysis. *SoftwareX* **10**, 100335 (2019)
21. Perret, B., Cousty, J.: Component tree loss function: Definition and optimization. In: *DGMM*. pp. 248–260 (2022)
22. Perret, B., Lefèvre, S., Collet, C., Slezak, É.: Hyperconnections and hierarchical representations for grayscale and multiband image processing. *IEEE Transactions on Image Processing* **21**, 14–27 (2012)
23. Ronse, C.: A topological characterization of thinning. *Theoretical Computer Science* **43**, 31–41 (1986)
24. Rosenfeld, A.: Adjacency in digital pictures. *Information and Control* **26**, 24–33 (1974)
25. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing* **7**, 555–570 (1998)
26. Salembier, P., Serra, J.: Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Transactions on Image Processing* **4**, 1153–1160 (1995)
27. Song, Y., Zhang, A.: Monotonic tree. In: *DGCI*. pp. 114–123 (2002)
28. Tao, R., Qiao, J.: Fast component tree computation for images of limited levels. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45(3)**, 3059–3071 (2023)