Binary Partition Tree Construction from Multiple Features for Image Segmentation

Jimmy Francky Tianatahina Randrianasoa, Camille Kurtz, Eric Desjardin, Nicolas Passat

Binary Partition Tree Construction from Multiple Features for Image Segmentation

Jimmy Francky Randrianasoa, Camille Kurtz, Éric Desjardin, Nicolas Passat

***Abstract*—In the context of digital image processing and analysis, the Binary Partition Tree (BPT) is a classical data-structure for the hierarchical modeling of images at different scales. BPTs belong both to the families of graph-based models and morphological hierarchies. They constitute an efficient way to define sets of nested partitions of image support, that further provide knowledge-guided reduced research spaces for optimization-based segmentation procedures. Basically, a BPT is built in a mono-feature way, *i.e.*, for one given image, and one given metric, by merging pairs of connected image regions that are similar in the induced feature space. We propose in this work a generalization of the BPT construction framework, allowing to consider versatile multi-feature paradigms. The cornerstone of our approach relies on a collaborative strategy enabling to establish a consensus between different metrics, thus allowing to obtain a unified hierarchical segmentation space. To reach that goal, we first revisit the BPT construction algorithm to describe it in a fully graph-based formalism. Then, we present the structural and algorithmic evolutions and impacts when embedding multiple features in BPT construction. We also discuss the different ways to tackle the induced memory and time complexity issues raised by this generalized framework. Final experiments illustrate how this multi-feature framework can be used to build BPTs from multiple images and / or multiple metrics computed through the image content.**

***Index Terms*—binary partition tree, morphological hierarchy, multiple features, graph-based image processing, image segmentation.**

I. Introduction

A. Context

In image processing and analysis, segmentation is a crucial task. The concept of segmentation is also quite generic from various points of views: in terms of semantics (from low-level definition of homogeneous areas to high-level extraction of specific objects), in terms of definition (object versus background or total partition of the image support), and in terms of algorithmics (region-based or contour-based approaches).

In this context, morphological hierarchies propose a wide range of data-structures for modeling images at various scales, allowing for the definition of connected operators [1]. Mainly based on the theoretical frameworks of graphs and mathematical morphology [2] [3, Chapters 3, 7, 9], these approaches have already proved their efficiency – the algorithms to build and handle them are generally of linear or quasi-linear time and space complexity – in many imaging applications. Their very principle is to embed images in a dual spatial / spectral representation space, composed of shapes (a shape is a spectrally homogeneous and spatially coherent region at a given scale) together with their spatial (neighbouring) and hierarchical (inclusion) relations. These representations offer a structured space to find the best regions / scales according to the applicative objective using, for instance, high-level features to describe the image regions and their content.

Among these representations, the Binary Partition Tree (BPT) [4] is a hierarchical representation of an image modeled as a tree structure, whose each node is a connected region. Each of these nodes is either a leaf – then corresponding to an elementary region – or an internal node, modeling the union of the regions of its two children nodes. The root is the node corresponding to the entire support of the image. Practically, a BPT is built from its leaves – provided by an initial partition of the image support – to its root, in a bottom-up fashion, by iteratively choosing and merging two adjacent regions which minimize a merging criterion (based on a given metric) computed between them. The BPT structure allows users to explore the image at different scales and can be used for various tasks such as segmentation, information retrieval, object recognition and visual browsing.

Like other hierarchical structures, the BPT was mainly designed to process one image at a time. In addition, by contrast with most of them (*e.g.*, component-trees, trees of shapes) that are intrinsically defined from the image content, the BPT is also designed to embed an extrinsic metric that is used – together with the image – to build a mixed image / knowledge model. In other words, a BPT is generally built for one image and one metric.

B. Motivations and Contributions

The BPT has already demonstrated its relevance for challenging image processing and analysis tasks, for instance in the fields of video, remote sensing or medical imaging. Nevertheless, as stated above, it remains mostly limited to a *one image, one metric* paradigm.

Indeed, on the one hand, from the imaging point of view, the handling of several images, *e.g.*, multisensor, multitemporal or multispectral, is generally dealt with either by defining a super-image, or by gathering the multiple information provided by various spectral bands into a single metric. On the other hand, the metric – namely the merging criterion used to decide, at each step of the BPT construction, which nodes to merge –

This work was supported by the French *Agence Nationale de la Recherche* under Grant ANR-10-BLAN-0205 and Grant ANR-12-MONU-0001.

J. F. Randrianasoa, É. Desjardin and N. Passat are with the Université de Reims Champagne Ardenne, CReSTIC, France (e-mail: {jimmy.randrianasoa,eric.desjardin,nicolas.passat}@univ-reims.fr).

C. Kurtz is with the Laboratoire d'Informatique Paris Descartes, Université Paris Descartes (Sorbonne Paris Cité), Paris 75006, France (e-mail: camille.kurtz@parisdescartes.fr).

is a scalar function, that imposes to fuse various elements of expert knowledge; this means in particular that this metric has to be carefully thought beforehand by the user, and cannot be handled on the flight by the process, in a more flexible fashion.

In this article, our purpose is to investigate how the BPT construction framework can be generalized to explicitly relax these two constraints, thus leading to a multi-feature paradigm, *i.e.*, the handling of *many images and / or many metrics*. The underlying idea of our approach relies on a collaborative strategy enabling to establish a consensus between different metrics, thus allowing to obtain as output a unified hierarchical segmentation space.

In order to reach that goal, it is first mandatory to emphasise the structural core of the BPT construction algorithm, and in particular to separate its fundamental graph-based expression – that is indeed a graph-collapsing algorithm – from its knowledge-based layers (image topology, metric definition, merging policies, etc.); this preliminary analysis is developed in Section III.

In a second time, we explain in Section IV, how the basic BPT construction framework can be generalized to handle multiple features. We first identify the requirements in terms of data-structures (Section IV-A), and then the algorithmic side effects (Section IV-B). A complexity analysis (Section IV-C) is then proposed to describe the induced space and time complexity increases.

In a third time, we discuss in Section V, how to practically tackle these complexity issues. To this end, we identify the main bottlenecks when considering standard sequential algorithmics (Section V-A). We then consider two points of view: first, heuristic solutions that approximate the exact algorithm in a sequential way (Section V-B); second, the way to switch from a sequential to a distributed algorithm (Section V-C).

This work is concluded by two application examples, in Section VI, that illustrate how the framework can allow us to define BPTs from multiple images and from multiple metrics.

The remainder of the article – which is an extended and improved version of the conference paper [5] – is composed of a synthetic state of the art of graph-based, hierarchical and multi-image segmentation, in Section II; and a conclusion that emphasises the perspectives of this work, in Section VII.

II. Related works

A. Graph-based and Hierarchical Image Segmentation

Image processing and analysis problems – and in particular segmentation – are often considered in a discrete way via concepts of graph theory. Practically, image points (*i.e.*, pixels, voxels) are considered as the vertices of a graph, while the spatial / neighbouring relations between them are modeled by the edges of this graph. This paradigm, democratized since the early 70's [6], led to the development of a wide range of segmentation approaches, based on basic graph manipulations.

In this context, image segmentation could be viewed as a partial (*e.g.*, region growing [7]), or total partitioning problem with partitions obtained via monotonic (*e.g.*, watersheds [8]) or non-monotonic transformations (*e.g.*, split and merge [9]). Some of these approaches led in particular to the development of optimization schemes (*e.g.*, graph-cuts [10], [11], random walks [12], power watersheds [13]). In the framework of mathematical morphology, these graph-based approaches gave rise to the notion of connected operators [14].

Graph-based segmentation allows us to obtain one segmentation result from a given image. In order to provide a better reliability to the ill-posed problem of segmentation, some hierarchical approaches were developed to compute families of nested partitions, providing potential solutions to segmentation issues at different scales. These notions of hierarchies for image analysis take their origin in image models initially devoted to optimize the access and space cost of the carried information (*e.g.*, octrees [15]). These regular models were progressively shifted toward image / content-guided, irregular hierarchies [16].

From this point on, several hierarchical image models were developed, mainly in the framework of mathematical morphology. The most popular are component-trees [17], trees of shapes [18], [19], hierarchical watersheds [20], hyperconnected component-trees [21], and binary partition trees [4] (see Section II-B). Since they provide a space of potential segmentations, instead of a single result, these hierarchical models were progressively involved in attribute-based [22] or optimization schemes [23], [24] for segmentation purpose.

Based on these image models, generally designed as trees (*i.e.*, rooted, connected, acyclic graphs), further developments were proposed to allow for a better flexibility in image and parameter handling. The case of multiband (for instance, colour) images was considered, leading to data-structures such as component-graphs [25], multivalued component-trees [26] or multivariate trees of shapes [27]. Topological handling was also investigated, by allowing connectivity hierarchies in component-hypertrees [28], or dealing with asymmetric hierarchies [29] allowing for non-directed graph as image models. The way to embed semantic information as image values was also pioneered via the notion of shaping [30], [31].

B. Binary Partition Trees

Most of the hierarchical structures proposed in the literature are models intrinsically deriving from the image signal. For instance, component-trees represent the inclusion of the successive level-sets; trees of shapes represent the image level-lines; while hierarchical watersheds rely on saliency measures similar to gradients. As a drawback, these models strongly rely on the image intensity since they aim to extract image regional extrema. Unfortunately, in different applicative contexts, such regions may not correspond to objects of interest in image content, in particular when dealing with complex images.

By contrast, the binary partition tree (BPT) [4] relies on a mixed image / knowledge model and allows for higher flexibility than many other hierarchical structures. From a structural point of view, BPTs present similar properties with binary space partition trees [32], designed to efficiently model an image space, mainly in computer graphics. Indeed, BPTs provide hierarchies of nested total partitions of an image.

From an algorithmic point of view, a BPT is built by progressively merging elementary image segments (for instance, flat zones), based on information about image content,

but also on a priori knowledge directly chosen with respect to the targeted segmentation application. In particular, the choice of the metric chosen to guide which pairs of regions should be iteratively fused together (and in which order) is crucial, since it can lead to a huge number of different BPTs, whose structures will be more or less adapted to certain kinds of tasks and images. This gave rise to several works, ranging from theoretical contributions [33] to experimental assessments [34]. The basic criteria used in most of image segmentation approaches are generally radiometric or geometric region similarities (or their fusion into a single metric). Thanks to this model, the BPT nodes are good candidates for capturing objects of interest, potentially emerging from the image content at different scales, since BPTs are able to build image regions based on their similarities.

Based on this property – and except few contributions in the field of object recognition [35], [36], [37] – BPTs were mainly involved in segmentation / classification cases where such total partitions make sense from a semantic point of view. More precisely, the wider application field of BPTs is remote sensing [38]. In this context, BPTs were involved for multiresolution / multiscale image segmentation and classification [39], [40]; coupled optical / LIDAR data analysis [41]; hyperspectral images [42], [43]; polarimetric SAR [44], [45], mixed SAR / hyperspectral images [46]; or SAR time series analysis [47].

C. Discussion – Contributions

The standard use of a single metric for the whole creation process of a BPT requires strong a priori knowledge from the user, regarding the size, the shape and / or the spectral homogeneity of the objects to be segmented in the image. For instance, the basic models and criteria used in most of image segmentation approaches are generally textural similarities [48], multi- / hyper-spectral homogeneity [42], or rely on combination of geometric and color criteria fused together in a single region similarity measure [39].

Such application-dependent strategy makes the process quite rigid and leads to a double issue. Firstly, the fusion of multiple features generally does not enable to take advantage of all the richness of the information carried by the diversity / complementarity of the metrics. In addition, in some specific cases, these merging criteria can be in disagreement together (*e.g.*, geometrical likeness vs. spectral dis-likeness) regarding the similarity between two adjacent regions, leading to a non-consensual BPT structure. Secondly, depending on the scale of the regions / nodes among the tree, some criteria may not be relevant for choosing the next couple of regions to be fused. For example, the closer the nodes are to the root (regions become wider), the less relevant a spectral homogeneity criterion is.

By contrast with the classical approach for building the BPT, this article proposes a new way of creating a BPT by using simultaneously various metrics. This paradigm also allows us, by definition, to encompass the case of processing several images / modalities of a same scene, with similar or specific metrics in each. In this work, we do not intend to regroup different criteria in a unique region merging criterion, but to consider each metric individually. The cornerstone of our framework relies on the consensual strategies – derived from the machine learning field – tuned for the management of these different complementary criteria.

Such framework presents some virtues. It lightens the task of the user, by offering more flexibility for the BPT creation. Indeed, since the negotiation between the different features, at each step of the BPT construction, is intrinsically dealt with by the algorithm, with respect to the chosen consensus policies, the hard prior knowledge mandatory from the user is reduced to the choice of these features and the global strategies for their collaboration.

The counterpart of these advantages is related to the necessary handling of more complex data-structures, and a higher computational cost. Thus, we devote a part of our study on setting up an adequate data-structure that allows the potential optimization of our algorithmic framework, and its distribution over several processing cores.

III. Structural Description of the BPT Construction

A. Definitions and Notations

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

An image is a function $I : \Omega \to V$ that associates to each point x of the finite set Ω a value $I(x)$ of the set V.

To model the fact that two points x and y of Ω are neighbours, let A_Ω be an adjacency (*i.e.*, irreflexive, symmetric) binary relation on Ω. In other words, $\mathfrak{G}_\Omega = (\Omega, A_\Omega)$ is a graph that models the structure of the image space.

For any partition $\mathcal{P}$ of Ω, we can define an adjacency inherited from that of Ω. More precisely, we say that two distinct sets $N_1, N_2 \in \mathcal{P}$ are adjacent if there exist $x_1 \in N_1$ and $x_2 \in N_2$ such that (x_1, x_2) is an edge of A_Ω, *i.e.*, x_1 and x_2 are adjacent in (Ω, A_Ω). This new adjacency relation $A_\mathcal{P}$ is also irreflexive and symmetric.

B. The Standard BPT Construction [4]

A BPT is a hierarchical representation of an image. More precisely, it is a binary tree whose each node is a connected region. A node can be either a leaf representing an "elementary" region, or a simple node modeling the union of two neighbouring regions. The root is the node corresponding to the support of the whole image.

The BPT is built in a bottom-up fashion starting from the determination of the leaves – provided by an initial partition of the image – to the root. This is done via an iterative process, that chooses and merges, at each step, two adjacent regions minimizing a criterion reflecting their likeness. This merging sequence is stored in a hierarchical structure which allows the regions of the image to be modeled at different scales. The BPT construction is illustrated in the right part of Figure 1.

A huge number of distinct BPTs may be obtained for a given initial partition of Ω. In order to decide which one among them will be the most relevant, it is necessary to define a *merging order*, *i.e.*, to decide of the priority of the fusions between nodes. Let $N_i, N_j \in \mathcal{P}$ be two distinct and adjacent

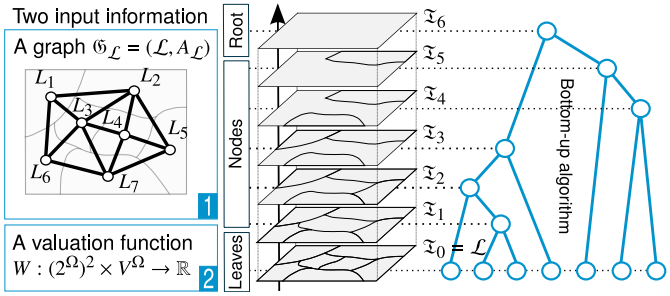


Fig. 1. Illustration of the algorithm for the creation of a BPT from one image. Left: two input information which are (1) a graph $\mathfrak{G}_\mathcal{L} = (\mathcal{L}, A_\mathcal{L})$ that models the initial partition of the image; and (2) a valuation function $W : (2^\Omega)^2 \times V^\Omega \to \mathbb{R}$ that allows us to choose, at each step of the process, the next pair of nodes to be merged (see Section III-C). Right: progressive, bottom-up, creation of the tree from $\mathfrak{T}_o$ to $\mathfrak{T}_6$ by iterative fusions of two neighbouring regions.

regions / nodes. A BPT generation relies on two main notions: a *region model*, denoted as $M_r(N_i)$, which specifies how a region N_i is characterized (*e.g.*, color, shape), and a *merging criterion*, denoted as $O_r(N_i, N_j)$, which defines the similarity of neighbouring regions N_i, N_j and thus the merging order.

A strategy commonly adopted to represent each region is to consider their average color in a specific color space (*e.g.*, RGB, HSV), and to iteratively merge pairs of adjacent regions that either have region models similar one to each other, or similar to the region model of the novel region built from their potential union. Another strategy [49] considers a region model based on a combination of radiometric and geometric features, with a merging criterion that weights both radiometric and geometric region similarities, evolving during the construction of the BPT, to provide a heavier weight to the geometric similarities according to the size of the BPT nodes. The choice for these parameters is strongly application-dependent. However, in any case, the merging criterion is a scalar function, which imposes to fuse various elements of expert knowledge, and cannot be handled on the flight by the process, in a more flexible and dynamic way.

C. Structural Description: A Graph-based Point of View

1) Graph and valuation function: The way to describe the construction of a BPT is generally considered from spatial (the way regions are built) and descriptive (the way regions are characterized and how they can be considered similar) points of view. Indeed, the classical – image and application-oriented – description of the BPT construction algorithm considers as input: the image I (*i.e.*, the geometrical embedding of Ω, and the value associated to each point of Ω); a region model, that allows us to "describe" the nodes; and a merging criterion, that allows us to quantify the homogeneousness of nodes before and after a putative fusion. These information are important from an applicative point of view.

However, beneath these image and knowledge-based notions, the construction of a BPT is intrinsically a process of graph collapsing. Indeed, from an algorithmic point of view, the only use of region models and merging criteria is to define a valuation on the edges that allows us to *choose* which nodes to fuse at any given step. In the sequel, we will then consider – without loss of correctness – that a BPT is fully[1] defined, by only two input information (see left part of Figure 1):

1) a graph $\mathfrak{G}_\mathcal{L} = (\mathcal{L}, A_\mathcal{L})$ that models the initial partition of the image;
2) a valuation function $W : (2^\Omega)^2 \times V^\Omega \to \mathbb{R}$ that allows us to choose (relatively to a specific metric), at each step of the process, the next pair of nodes to be merged.

2) Structural description of the algorithm: Now, let us consider an initial partition $\mathcal{L}$ of Ω. (Each node $L \subseteq \Omega$ of $\mathcal{L}$ is generally assumed to be connected with respect to A_Ω.) This partition $\mathcal{L}$ defines the set of the leaves of the BPT we are going to build (*e.g.*, $\mathcal{L}$ can be the set of the image flat zones). It is equipped by the adjacency $A_\mathcal{L}$ inherited from A_Ω, leading to a graph $\mathfrak{G}_\mathcal{L} = (\mathcal{L}, A_\mathcal{L})$ that models the structure of the partition of the image I.

The BPT is the data-structure that describes the progressive collapse of $\mathfrak{G}_\mathcal{L}$ onto the trivial graph $(\Omega, \emptyset)$. This process consists of defining a sequence $(\mathfrak{G}_i = (\mathcal{N}_i, A_{\mathcal{N}_i}))_{i=0}^n$ (with $n = |\mathcal{L}| - 1$) as follows. First, we set $\mathfrak{G}_0 = \mathfrak{G}_\mathcal{L}$. Then, for each i from 1 to n, we choose the two nodes N_{i-1} and N'_{i-1} of $\mathfrak{G}_{i-1}$ linked by the edge $(N_{i-1}, N'_{i-1}) \in A_{\mathcal{N}_{i-1}}$ that minimizes the valuation function W, and we define $\mathfrak{G}_i$ such that $\mathcal{N}_i = \mathcal{N}_{i-1} \setminus \{N_{i-1}, N'_{i-1}\} \cup \{N_{i-1} \cup N'_{i-1}\}$; in other words, we replace these two nodes by their union. The adjacency $A_{\mathcal{N}_i}$ is defined accordingly from $A_{\mathcal{N}_{i-1}}$: we remove the edge (N_{i-1}, N'_{i-1}), and we replace each edge (N_{i-1}, N''_{i-1}) and / or (N'_{i-1}, N''_{i-1}) by an edge $(N_{i-1} \cup N'_{i-1}, N''_{i-1})$ (in particular, two former edges may be fused into a single).

From a structural point of view, the BPT $\mathfrak{T}$ is the Hasse diagram of the partially ordered set $(\bigcup_{i=0}^n \mathcal{N}_i, \subseteq)$. From an algorithmic point of view[2], $\mathfrak{T}$ is built in parallel to the progressive collapse from $\mathfrak{G}_0$ to $\mathfrak{G}_n$; in other words, $\mathfrak{T}$ stores the node fusion history. More precisely, we define a sequence $(\mathfrak{T}_i)_{i=0}^n$ as follows. We set $\mathfrak{T}_0 = (\mathcal{N}_0, \emptyset) = (\mathcal{L}, \emptyset)$. Then, for each i from 1 to n, we build $\mathfrak{T}_i$ from $\mathfrak{T}_{i-1}$ by adding the new node $N_{i-1} \cup N'_{i-1}$, and the two edges $(N_{i-1} \cup N'_{i-1}, N_{i-1})$ and $(N_{i-1} \cup N'_{i-1}, N'_{i-1})$. The BPT $\mathfrak{T}$ is then defined as $\mathfrak{T}_n$.

3) Data-structures: The above description of the BPT construction algorithm implies to define – and update during the whole process – several data-structures, namely:

- the graph $\mathfrak{G}$, that allows us to know what nodes remain to be merged and what are their adjacency links; and
- the tree $\mathfrak{T}$ that is progressively built.

In order to efficiently compute the valuation W, it is also important to associate each node of $\mathfrak{G}$ to the corresponding part of the image I, *e.g.*, via a mapping between $\mathfrak{G}$ and Ω.

The last – but not least – required data-structure is a sorted list $\mathcal{W}$ that gathers the scalar valuations of each remaining edge of $\mathfrak{G}$. This list contains the information that will authorise, at each of the n iterative steps of the process, to choose

[1]The construction of a BPT is not fully deterministic since it may happen that the valuation function W has a common minimal value for several edges.

[2]In [50], a graph-based definition of BPT construction is also proposed, that relies on a minimum spanning tree paradigm. However, this formalization is valid only if the merging order is associated to a valuation of the edges that is fixed and defined *a priori* on the initial partition.

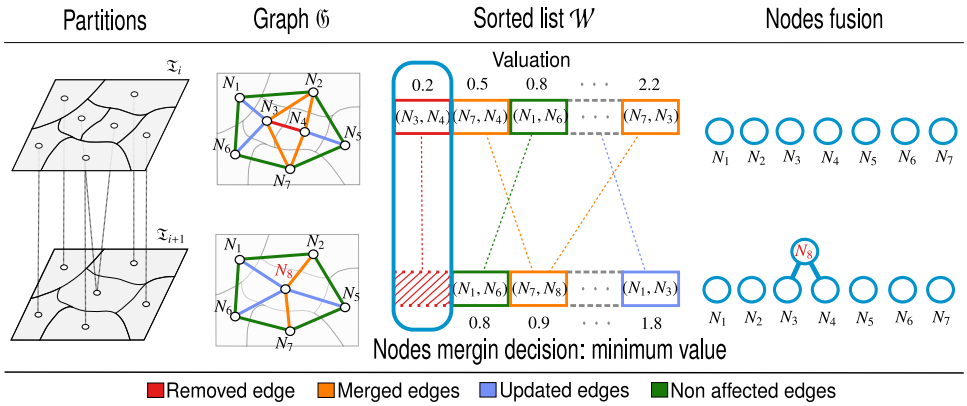


Fig. 2. One step of the building of a BPT from one image. From the left to the right: First: the partition of Ω before and after the fusion of two nodes. Second: the associated graph $\mathfrak{G}$, before and after the fusion of N_3 and N_4, forming the new node N_8; the red edge is removed; the blue and orange edges are updated, *e.g.*, (N_1, N_3) becomes (N_1, N_8); the orange are merged by pairs, *e.g.*, (N_7, N_3) and (N_7, N_4) become (N_7, N_8); the green edges are not affected. Third: the sorted list that gathers the scalar valuations of each remaining edge of $\mathfrak{G}$; the red cells are removed, as the edge (N_3, N_4) is suppressed; this edge had been chosen due to its highest position in the list; the scores of blue and orange cells are updated with respect to N_8; the orange cells are merged by pairs; the positions of the blue and orange cells are updated with respect to their new scores; the scores of the green cells are not affected. Last: a new part of the BPT $\mathfrak{T}$ is created by adding the new node N_8, and linking it to its two children nodes N_3 and N_4.

the couple of nodes to be merged relatively to a given metric. One iteration of this algorithm is illustrated in Figure 2.

This choice is made in constant time $\mathcal{O}(1)$, since $\mathcal{W}$ is sorted. After the merging operation, $\mathcal{W}$ has to be updated: (1) to remove the edge between the two nodes; (2) to update the edges affected by the merging operation; and (3) to re-order these updated edges. Operation (1) is carried out in constant time $\mathcal{O}(1)$. Operation (2) is carried out in $\mathcal{O}(\alpha.T_W)$, where T_W is the cost of the computation of W for an edge, and α is the number of neighbours of the merged nodes (α is generally bounded by a low constant value). Operation (3) is carried out in $\mathcal{O}(\alpha.\log_2|\mathcal{W}|)$.

IV. Multiple Feature Generalization of the BPT Construction

In this section, we investigate a multi-feature generalization of the BPT construction algorithm. Indeed, we now consider that this construction, viewed as a graph collapsing problem, still takes as input the graph $\mathfrak{G}_\mathcal{L} = (\mathcal{L}, A_\mathcal{L})$. By contrast, we now consider *several* valuation functions $W_\star : (2^\Omega)^2 \times V^\Omega \to \mathbb{R}$. These functions are still devoted to allow us to choose, at each step of the process, the next pair of nodes to be merged. We discuss the structural and algorithmic side effects of using several valuation functions "at the same time", instead of only one. Our purpose is still to build one BPT from these input information. Practically, introducing several valuation functions allows us to model and embed several features in an independent way in the construction process. These features can in particular represent several metrics associated to a same image; a same metric associated to several images of a same scene; or even various metrics on various images of a same scene. In other words, this generalized paradigm opens the way to a versatile handling of multi-image (*e.g.*, multimodal, multi-time, etc.) and / or multi-criteria definition of consensual BPTs, without the constraint of defining beforehand any ad hoc, hard, unified metric.

A. Structural Evolutions

The proposed generalization deals with the "feature" part of the construction. As stated in Section III-C1, we need a graph that models the initial partition $\mathcal{L}$ of the image(s). Here, we still have one such initial graph; which means that – fundamentally – our purpose is still to collapse a unique graph, while – practically – our purpose is to build the BPT associated to a unique spatial scene, topologically modeled by this graph. From a semantic point of view, this implies that the (potentially multiple) images involved in the BPT construction process have to be defined in a same spatial reference[3], *i.e.*, the same support Ω. A graph $\mathfrak{G}_\mathcal{L}$ – which is isomorphic to (Ω, A_Ω) – can be obtained easily, either by subdividing Ω into one-point singleton sets or by considering flat zones.

The "graph" part of the BPT construction process thus remains unchanged. In terms of data-structures, the generalized BPT construction algorithm will still handle one graph $\mathfrak{G}$, that will be progressively collapsed; and one tree $\mathfrak{T}$ that will be built to finally provide the BPT. A unique mapping between $\mathcal{N}$ and Ω will still allow us to have access to the values of a node for the different images.

Let us now consider the "feature" part of the data-structure. In the initial BPT construction approach, the valuation function $W : (2^\Omega)^2 \times V^\Omega \to \mathbb{R}$ was explicitly modeled by a sorted list $\mathcal{W}$ of the values of all edges of the graph. This list was updated during the progressive collapsing of $\mathfrak{G}$, by removing elements from the list; updating the values of some edges (in particular, thanks to the mapping between $\mathcal{N}$ and Ω); and (re)sorting edges with respect to their updated values.

We now consider $n > 1$ valuation functions $W_\star : (2^\Omega)^2 \times V^\Omega \to \mathbb{R}$, which means that each edge is associated to n values, one for each function. For instance, by assuming that we consider k distinct images, and l distinct metrics, we may

[3]This constraint is not actually a real issue, since this spatial coherence assumption is generally a standard requirement in image processing (*e.g.*, in medical imaging where several modalities are superimposed via registration processes; in remote sensing where the acquired data are georeferenced; etc.).

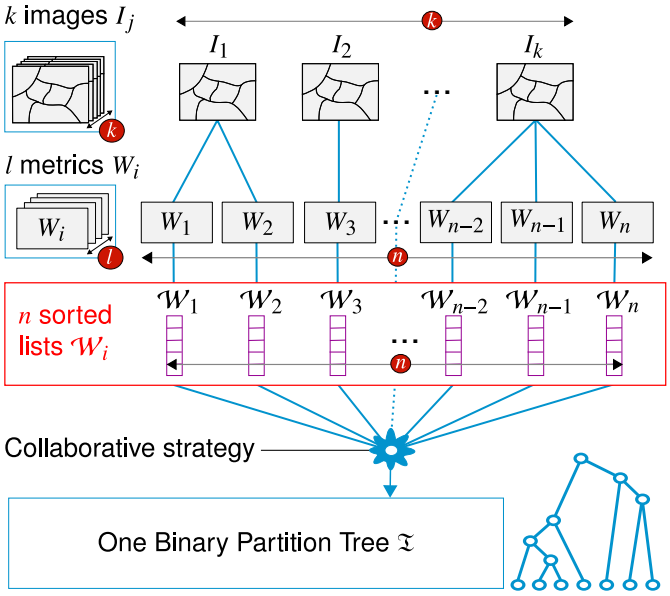


Fig. 3. General structure of a BPT creation involving n valuation functions / sorted lists. Each sorted list $\mathcal{W}_i$ gathers edges information computed by using one valuation function corresponding to a given metric on a given image. A BPT is obtained by applying a collaborative strategy leading to a consensus between the information carried by these n sorted lists (see Section IV-B).

have up to $n = k.l$ such valuation functions. This leads us to define no longer one, but n sorted lists $\mathcal{W}_i$ $(1 \leq i \leq n)$. Each sorted list $\mathcal{W}_i$ is associated with a specific valuation function $W_i : (2^\Omega)^2 \times V_j^\Omega \to \mathbb{R}$ that is defined in particular with respect to a value set V_i (see Figure 3). The handling of these sorted lists remains the same in terms of removal, value updating / resorting, as for one list.

Our purpose is now to build a BPT from these n lists, by generalizing the algorithmic framework described in Section III-B, which initially depended on one list $\mathcal{W}$.

B. Algorithmic Evolutions

From an algorithmic point of view, each iteration of the construction process preserves the same organization. An edge is chosen and the two incident nodes of the graph are merged. This operation leads to update the nodes and edges of $\mathfrak{G}$, and adds a new node plus two edges in $\mathfrak{T}$. The main differences are that: (1) $n > 1$ sorted lists then have to be updated; and (2) the choice of the optimal edge has to be made with respect to the information carried by these n sorted lists instead of only one, for a standard BPT.

At each iteration of the algorithm (see Figure 4), the choice of the optimal edge to remove, leading to the fusion of its two incident nodes, depends on a decision, *i.e.*, a consensus, made with respect to the information provided by these n lists. In particular, useful information are carried, on the one hand, by the sorted lists $\mathcal{W}_i$, that give a *relative* information on edges, induced by their ordering with respect to W_i; on the other hand, by the valuation functions $W_i : (2^\Omega)^2 \times V_j^\Omega \to \mathbb{R}$ that give an *absolute* value to each edge. These information are of distinct natures; we study their relevance according to various kinds of consensus policies. In particular, we identify, hereinafter, three main families of consensus strategies.

1) Absolute information consensus: Let us consider that the consensus policy consists of choosing the edge of lowest mean valuation among the n lists $\mathcal{W}_i$, or the edge of minimal valuation among all lists. The first consensus (namely *min of mean*) is defined by a linear formulation: $\arg_{(N,N')\in\mathcal{N}} \min \sum_{i=1}^{n} W_i((N,N'))$, while the second (namely *min of min*) is defined by a non-linear formulation: $\arg_{(N,N')\in\mathcal{N}} \min \min_{i=1}^{n} W_i((N,N'))$. In both cases the decision is made by considering the *absolute* information carried by the edges. In such conditions – and more generally whenever the information carried by the values of each edge is a sufficient knowledge, independently of the relative values between edges – n sorted lists $\mathcal{W}_i$ are not necessary, and a single sorted list $\mathcal{W}$ that contains the information of these, linear or non-linear, formulations is indeed sufficient. The BPT construction involving n lists is then equivalent to that from one list.

The main difficulty raised by this policy derives from the potential heterogeneity of the values carried by the different $W_\star$ valuation functions. Indeed, to be tractable, this policy requires that all values are comparable. This implies in particular that they must be of same nature, but also that they should be normalized to allow for the definition of adequate fusion / comparison operators. This issue is mainly the same that occurs in most optimization problems where a given metric is built from several terms of varying semantics. This drawback argues in favour of using the next two proposed policies.

2) Relative local information consensus: Let us now consider that the consensus policy consists of choosing the edge that is the most often in first position in the n sorted lists $\mathcal{W}_i$, or the most frequently present in the $r \ll |\mathcal{W}_i|$ first positions in the n sorted lists $\mathcal{W}_i$. These consensus (namely, *majority vote* and *most frequent*, potentially weighted) policies do not act on the absolute valuations of the edges, but on their relative positions in the lists. In such case, it is then mandatory to maintain n sorted lists. However, the decision process does not require to explicitly access the whole lists, but it can be restricted to the first (or the first r) element(s) of each, leading to a *local* decision process.

3) Relative global information consensus: Let us finally consider that the consensus policy consists of choosing the edge that has the best global ranking among the n sorted lists $\mathcal{W}_i$. As previously, such consensus (*e.g.*, *best average*, or *best median ranking*) policy, also acts on the relative positions of the edges in the lists, and does not need to consider the absolute values of the edges. However, by contrast with the above case, the decision process requires to explicitly access the whole content of all these lists, leading to a *global* decision process of higher computational cost.

C. Complexity Analysis

Computationally, choosing the edge to remove is no longer a constant time operation, but will depend on the way information are used and compared. Afterwards, operations (1–3) described in the standard BPT construction algorithm, for the

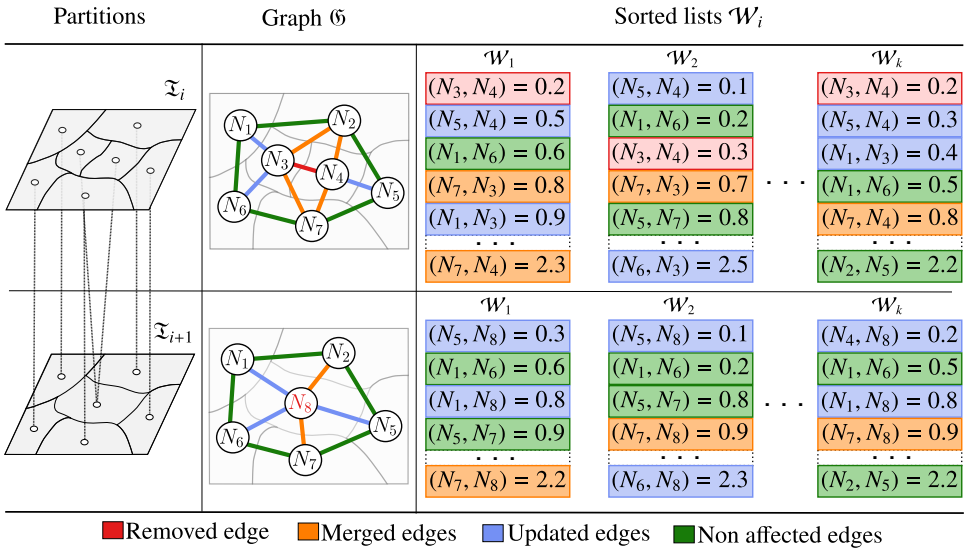


Fig. 4. One step of the building of a BPT involving n lists $\mathcal{W}_i$. Left: the partition of Ω before and after the fusion of two nodes. Center: the associated graph $\mathfrak{G}$, before and after the fusion of N_3 and N_4, forming the new node N_8; the red edge is removed; the blue and orange edges are updated, *e.g.*, (N_1, N_3) becomes (N_1, N_8); the orange edges are merged by pairs, *e.g.*, (N_7, N_3) and (N_7, N_4) become (N_7, N_8); the green edges are not affected. Right: the n lists $\mathcal{W}_i$, each corresponding to a valuation function W_i; the red cells are removed, as the edge (N_3, N_4) is suppressed; this edge had been chosen according to a given consensus policy, due to its "optimal" position and / or valuation in the n lists; the scores of blue and orange cells are updated with respect to N_8; the orange cells are merged by pairs; the positions of the blue and orange cells are updated with respect to their new scores; the scores of the green cells are not affected.

sorted list maintenance, have to be duplicated for each list. These operations are then carried out in $\mathcal{O}(n)$, $\mathcal{O}(n.\alpha.T_{W_\star})$ and $\mathcal{O}(n.\alpha.\log_2|\mathcal{W}_\star|)$, respectively, where T_{W_i} is the cost for computing W_i for a given edge, while α is an upper bound for the nodes degree within the graph $\mathfrak{G}$.

However, this initial generalization of the BPT construction algorithm can be refined by studying more precisely the policies that are considered to choose an edge, with respect to the information carried by the valuation functions W_i and / or the sorted lists $\mathcal{W}_i$.

Besides, the choice of a consensus strategy is strongly application-dependent. As a consequence, it is important to consider a trade-off between the structural and computational cost of the approach versus the benefits in terms of results accuracy. In particular, these costs are summarized in Table I.

This table provides the cost of an elementary step of the BPT construction process. The number of these steps is equal to the size of the initial graph $\mathfrak{G}_\mathcal{L}$, namely $|\mathcal{L}|$, and more precisely to the number of vertices in this initial graph (minus one), as each step merges two of these vertices, until obtaining a graph formed by exactly one node. At each step, at least one edge is removed from the graph; the number of remaining edges is in particular equal to the size of the list(s) $|\mathcal{W}_i|$. We can assume that the number of edges is bounded by the number of vertices of the graph, up to a multiplicative constant α (generally low for images defined on discrete grids, 4 in general on pixel 2D images). Based on these assumptions, an upper bound for the overall computational cost of the standard BPT construction [4] is $\sum_{i=1}^{|\mathcal{L}|}(\mathcal{O}(1)+\mathcal{O}(\log_2|\mathcal{W}|)) = \sum_{i=1}^{|\mathcal{L}|}\mathcal{O}(1) + \sum_{i=1}^{|\mathcal{L}|}\mathcal{O}(\log_2(\alpha.i)) = \mathcal{O}(|\mathcal{L}|) + \log\alpha.\mathcal{O}(\log_2(|\mathcal{L}|!)) = \mathcal{O}(|\mathcal{L}|\log_2|\mathcal{L}|)$, and so is the cost for the first consensus policy (Section IV-B1). Following the same kind of calculation, the cost for the second consensus policy (Section IV-B2) is $\mathcal{O}(n.|\mathcal{L}|\log_2|\mathcal{L}|)$, while that of the third (Section IV-B3) is $\mathcal{O}(n.|\mathcal{L}|^2)$.

V. Scaling up the Multi-feature BPT Construction

A. Bottlenecks

The above complexity analysis emphasises three main bottlenecks for the computational cost of BPTs in general, and in particular for multi-feature BPTs. These bottlenecks are:

(1) the size of the edge lists, that is linearly dependent on the size $|\mathcal{L}|$ of the initial graph, which leads to the $\log_2|\mathcal{L}|$ part of the cost (and a $|\mathcal{L}|$ part for the third policy);
(2) the number of steps of the construction process, that is linearly dependent on the size $|\mathcal{L}|$ of the initial graph, which leads to the $|\mathcal{L}|$ part of the cost;
(3) the number of valuation functions, which leads to the n part of the cost.

We discuss hereinafter some sequential and distributed strategies devoted to decrease these bottleneck parameters, thus leading to algorithms that will lead to fairly similar results as the exhaustive algorithm, but with a lower overall complexity.

B. Heuristics for Sequential Algorithmics

The bottleneck deriving from the number of valuation functions can be hardly avoided when considering a sequential algorithmics. In such case, the optimization of the framework mainly relies on heuristic strategies designed for standard mono-feature BPT construction; they can of course also allow us to decrease that of multi-feature BPTs.

The first way to reduce the computational complexity of the BPT construction consists of initiating the process from a

TABLE I
COST OF THE BPT CONSTRUCTION FOR VARIOUS FAMILIES OF CONSENSUS POLICIES. FOR THE SAKE OF READABILITY, α AND $T_{W_\star}$, WHICH ARE PRACTICALLY BOUNDED BY LOW CONSTANT VALUES HAVE BEEN OMITTED HERE.

	# $\mathcal{W}_\star$	Edge choice	Edge removal	Edges update	Edges sorting
Standard BPT construction [4]	1	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(\log_2 \lvert\mathcal{W}\rvert)$
Absolute inf. (Section IV-B1)	1	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(\log_2 \lvert\mathcal{W}\rvert)$
Relative local inf. (Section IV-B2)	n	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n.\log_2 \lvert\mathcal{W}_\star\rvert)$
Relative global inf. (Section IV-B3)	n	$\mathcal{O}(n.\lvert\mathcal{W}_\star\rvert)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n.\log_2 \lvert\mathcal{W}_\star\rvert)$

partition $\mathcal{L}$ with a lower number of regions. Instead of using singleton sets, which is equivalent to setting $\mathcal{L}$ isomorphic to Ω, and thus of same cardinality, flat zones or superpixels [51] can be considered. In such case, the gain of complexity derives from the reduction of $|\mathcal{L}|$ with the counterpart of adding a cost for flat zones (or superpixels) computation.

Another solution to optimize the overall cost is to reduce the cost of each elementary step. In particular, the update of the edges – and more precisely their resorting within the list(s) – after node fusions, can be carried out only after a given number ρ of iterations. This optimization is however marginal, since these edges still have to be resorted, while the gain concerns the lower size of the list at the resorting time. This optimization only influences the $\log_2 |\mathcal{L}|$ part of the computational complexity. In addition, the risks of choosing non-optimal edges for the merging, increase linearly with ρ.

An alternative consists of working with lists that do not contain all the edges [52]. This heuristic relies on decreasing the number of edges by only adding in the list(s) those of lower values during each update stage. This strategy leads to work on lists of lower complexities, acting on the $|\mathcal{L}| \log_2 |\mathcal{L}|$ term of the cost. As the previous one, this algorithm does not guarantee that the BPT structure will be topologically equivalent to the one of a BPT built using the original algorithm.

C. From Sequential to Distributed Algorithmics

As observed above, the sequential optimizations of the mono-feature BPT construction algorithm necessarily affect the result, by providing an approximate BPT, compared to the original algorithm. In addition, such heuristics do not tackle the issue of efficiently handling multiple lists, in the case of multi-feature BPT construction. In this context, we explore strategies based on distributed algorithmics that enable to build – approximate but fairly close – BPTs and that will allow us to effectively break the complexity of the initial algorithm.

The time cost of the multi-feature BPT construction is mainly due to the space cost of the handled data-structures, and in particular the number (n) and size ($|\mathcal{L}|$) of the lists $\mathcal{W}_i$. To reduce this time cost via distributed algorithmic paradigms, it is mandatory to split this space cost, by partitioning the amount of information to be stored and processed. In this context, two alternatives can be considered: either distributing each whole list on a given computing core; or splitting the graph $\mathfrak{G}$ to be processed and distributing each subgraph on these cores.

1) List-based distribution algorithmics: By assuming that n cores can be used, one list $\mathcal{W}_i$ can be assigned to each core. In such case, each core is able to process – in parallel and without interaction with the others – some list-dependent operations such that the removal of an edge from the list; the update of the edges impacted by the fusion of the two vertices incident to the removed edge; and the re-sorting of the lists after these updates. The speed-up for these operations is then linear, and we preserve the same cost as for standard BPT construction for these parts of the building process.

However, the choice of the edge to be removed at each step still requires to exchange information between the n lists / cores. Then the distribution of the lists has no speed-up effects on this part of the process.

In summary, the properties, advantages, and drawbacks of this distribution strategy are the following:

- the number of algorithmic cores is constant and determined a priori as the number n of lists;
- only the lists are distributed on the cores, while the graph and image structures are shared;
- the distributed algorithm is equivalent to the sequential one in terms of result;
- the time complexity of the overall process remains bounded by a n factor, due to the communications between lists for choosing the best edge at each step;
- the time cost is however reduced due to the linear speed-up of the other operations on lists.

2) Graph-based distribution algorithmics: The dual solution to distributing one list on each algorithmic core, is to split the image space into p sets of nodes, and to distribute each node set – and the associated edges – onto p cores. In such condition, each core still has to handle n lists $\mathcal{W}_i$, but each of these lists can be restricted only to the edges of the associated sub-image / sub-graph. Then, the spatial coherence of each split sub-image / sub-graph is crucial; indeed, at each step, the removal of an edge due to the fusion of two nodes, implies to update and resort the edges in a direct neighbourhood. As a consequence, a regular subdivision of the image into squares (or via superpixels) is mandatory.

This space-partitioning strategy also implies to deal with the case of the edges that are shared by two sub-images, *i.e.*, the edges whose the two vertices are located in different sub-graphs, respectively. The handling of these edges is directly linked to the policy for handling the evolution of the partition, during the iterative construction of the BPT. Indeed, each core processes its own edges, within its own subgraph. In other words, each core builds a BPT for its handled sub-image, independently from the other cores. This modus operandi is a fair approximation, in terms of results, of what should be obtained with a sequential algorithm. This assertion is notably

justified when the distribution of the W_i values of edges is homogeneous over the image, thus ensuring that each sub-image contains similar values at a same step.

In particular, this is verified for sub-images that are correctly designed with respect to the evolution of the BPT construction. More precisely, it is important to progressively fuse the sub-images and their associated partial BPTs, and to carry out these merging just in time. In this context, two approaches can be considered. The first is deterministic and parametrised beforehand, by experimentally assessing the number of iterations required before each merging of a regular (*e.g.*, quad- / octree) hierarchical partitioning, to approximate at best a sequential BPT construction by a distributed one. The second is non-deterministic, and consists of merging two subgraphs whenever an edge that links these two sub-graphs presents a W_i value that is lower that any edge in both sub-graphs.

In summary, the properties, advantages, and drawbacks of this distribution strategy are the following:

- the number of algorithmic cores is scalable, and requires to be sufficiently high to handle a hierarchical decomposition into sub-images;
- each one of the p core contains n lists, but their size is p times lower than in the sequential case;
- each core contains the subgraph it handles;
- the distributed algorithm is not equivalent to the sequential one in terms of result, as some merging operations are carried out in parallel and without communication; however the result can be designed to be similar to the one obtained with a sequential approach;
- the speed-up of the process is directly correlated with the parallelization degree over the hierarchy of subgraphs.

VI. Application Examples

To illustrate our framework, two application cases have been considered, respectively in the domain of remote sensing and medical imaging. In Section VI-A, we show how a multi-feature BPT can be built from a single satellite image by considering simultaneously, and in a consensual manner, various metrics. In Section VI-B, we show how a multi-feature BPT can be built from radiological images by considering simultaneously information provided by multimodal images. Our purpose is to highlight the versatility of the multi-feature framework by demonstrating how it can be used to build BPTs from either multiple images and / or multiple metrics computed through the image content.

For the sake of reproducibility, we developed a JAVA library[4] that implements the multi-feature BPT creation algorithm. In this library, each BPT can be interactively browsed, in a threshold-like fashion, thus enabling to easily determine the desired segmentation (globally, and / or by refining one or several branches). We have also developed and integrated in this software, a tiff library which allows to load only the subdivisions of the images that are necessary to the current segmentations. This enables to reduce the memory resources required by the application, which is useful when dealing with large images.

[4]Available at https://bitbucket.org/agat-team/agat-application.

The BPT construction and segmentation approaches were voluntarily chosen as very simple, in order to avoid any bias related to these choices, thus better focusing on the actual structural effects of multi-feature BPT versus standard BPT. These experiments then have to be considered as illustrative examples, since neither quantitative validation nor fine parameter tuning were carried out. Our purpose is mainly to give the intuition of potential uses of such BPTs in various complex imaging domains.

A. Illustrative example 1: Multi-criteria segmentation

The segmentation of very-high spatial resolution (VHSR) satellite images is a challenging task since the latest generation of images presents high spectral and spatial resolution properties, leading to huge volumes of data. In this context, the segmentation of satellite images using classical mono-metric BPTs has already been widely studied [38], [39], [40], [42], [43]. This motivates in particular the experimentation of multi-criteria segmentation procedures on such kinds of images.

1) Data: The dataset used here (courtesy LIVE, UMR CNRS 7263) was sensed over the town of Strasbourg (France). The original sample is an urban VHSR image (1024×1024 pixels) acquired by the PLÉIADES satellite in 2012 (Figure 5(a–d)). It is a pansharpened image at a spatial resolution of 60 cm with four spectral bands (NIR, R, G, B). A ground-truth map of the urban objects represented in the scene is also available (Figure 5(e–h)).

2) Method and results: To reduce the spatial complexity of this approach, the BPTs are built from an initial partition $\mathcal{L}$ composed of 4484 regions obtained by applying a Mean-Shift clustering (spatial radius $= 30$, spectral value $= 10$, minimum region size $= 100$). We considered here various valuation functions $W_\star : (2^\Omega)^2 \times V^\Omega \to \mathbb{R}$:

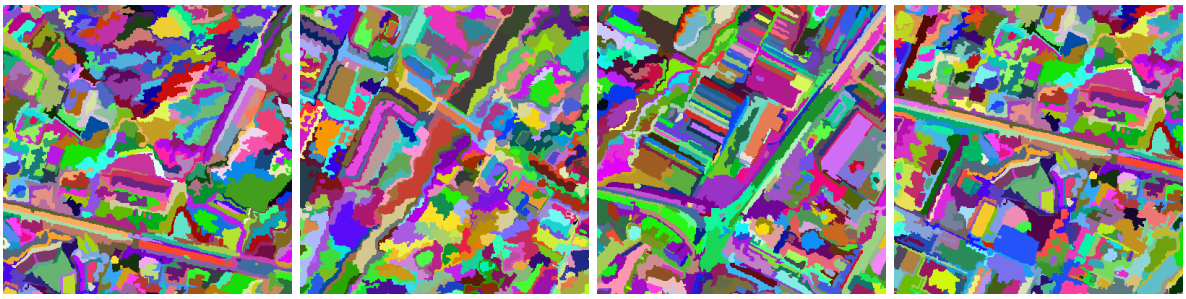
- the increase of the ranges of the intensity values (for each spectral band);
- the change of region elongation values;
- the change of region smoothness values;

induced by the fusion of the regions. In the case of multi-criteria BPTs, the relative local information consensus policy *mean-of-ranks*, according to the position of the edges within the lists is applied for the first 20% of the lists $\mathcal{W}_\star$. We only consider the intensity value criterion for building the "standard" BPT.

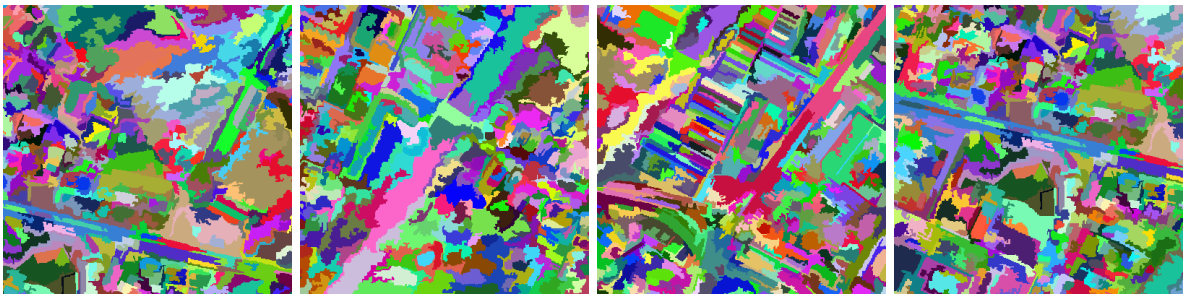
The "standard" BPT of the satellite image of Figure 5(a–d) is first segmented by considering a user-defined horizontal cut to produce the same number of regions as in the ground-truth map, see Figure 5(i–l). The multi-criteria BPT is then segmented in the same way, leading to the same number of regions, see Figure 5(m–t). For visualisation purpose, the segmentation results are depicted here in random false colours.

From the "standard" BPT result, we observe that the obtained regions are quite radiometrically homogeneous and are well adapted to extract simple urban objects (*e.g.*, small house roofs, forest and vegetation areas). Concerning more complex urban objects, which are strongly structured by their geometrical shapes (*e.g.*, rivers and roads with elongated structures), the regions produced with the "standard" BPT

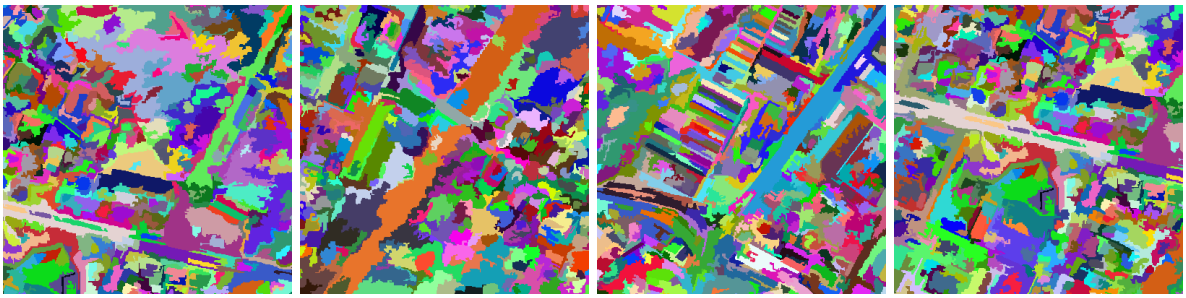
(a) VHSR image crop (1). (b) VHSR image crop (2). (c) VHSR image crop (3). (d) VHSR image crop (4).



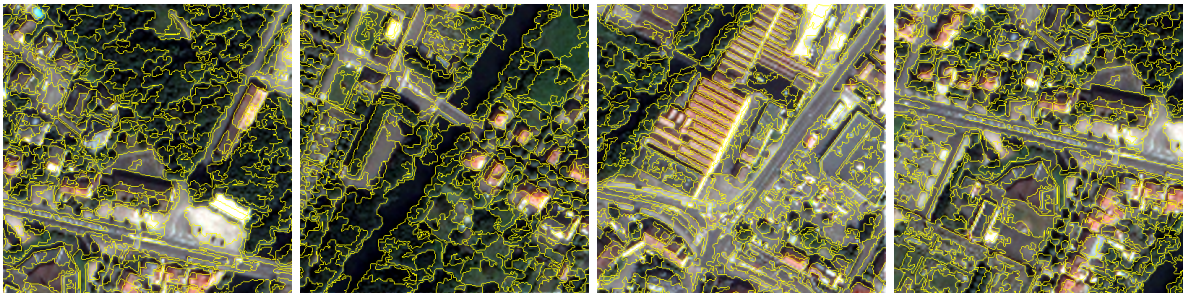
(e) Ground truth crop (1). (f) Ground truth crop (2). (g) Ground truth crop (3). (h) Ground truth crop (4).



(i) Standard BPT result (1). (j) Standard BPT result (2). (k) Standard BPT result (3). (l) Standard BPT result (4).

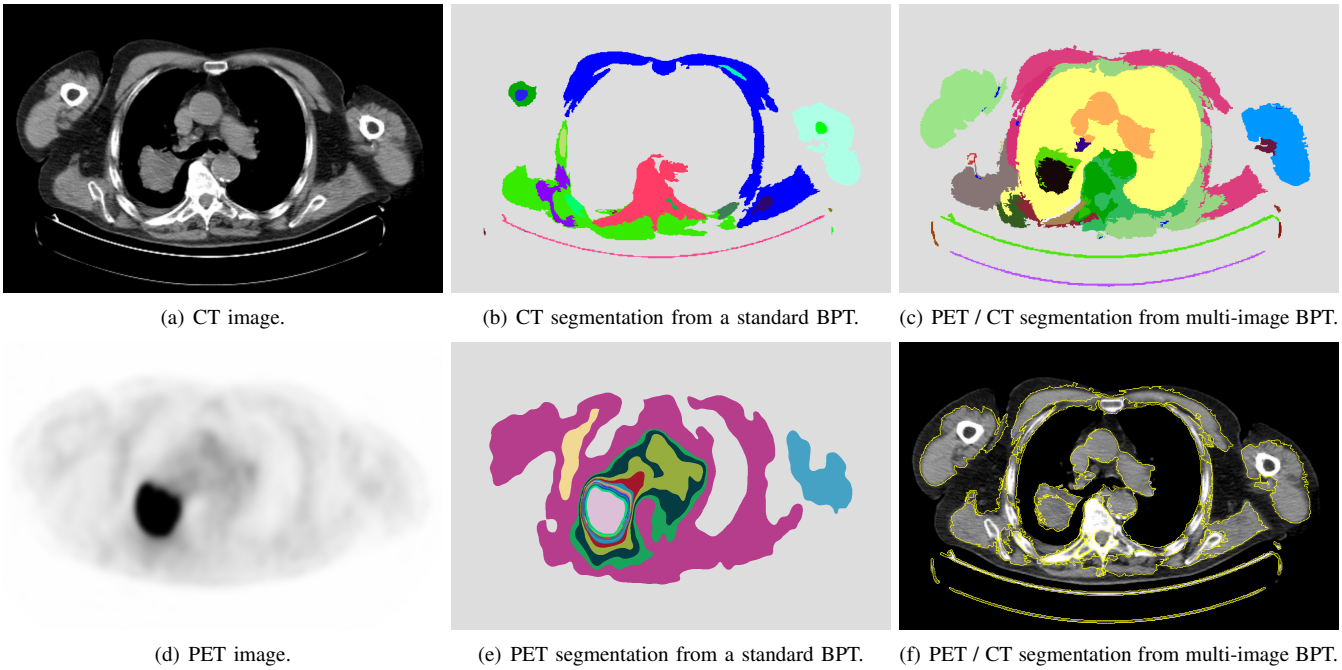


(m) Multi-criteria BPT result (1). (n) Multi-criteria BPT result (2). (o) Multi-criteria BPT result (3). (p) Multi-criteria BPT result (4).



(q) Boundary image of (m). (r) Boundary image of (n). (s) Boundary image of (o). (t) Boundary image of (p).

Fig. 5. Segmentation of very-high spatial resolution (VHSR) satellite images, see Section VI-A.



(a) CT image. (b) CT segmentation from a standard BPT. (c) PET / CT segmentation from multi-image BPT.

(d) PET image. (e) PET segmentation from a standard BPT. (f) PET / CT segmentation from multi-image BPT.

Fig. 6. Multi-image segmentation in the context of radiological multimodal PET / CT imaging.

are not relevant since the considered urban objects are often composed of several small sub-regions (see the river in Figure 5(j)). In comparison, the cut extracted from the multi-criteria BPT enables to directly gather, in a same partition, regions corresponding both to simple urban objects and to complex ones (see the house roofs on top of Figure 5(o) and river and road sections in Figure 5(o,p)). Our assumption is that complex urban objects appearing in the image content as homogeneous and elongated can be extracted from a cut of the multi-criteria BPT thanks to the consensus made between the different criteria during the BPT creation. However, some complex urban objects, represented as a single segment in the ground-truth image (see the curved road part on left bottom of Figure 5(g)) are still divided in the partition obtained with the multi-criteria BPT (see Figure 5(o)). This is probably due to the definition of the elongation criterion implemented as the ratio between the bounding box sides of the segment.

B. Illustrative example 2: Multi-image segmentation

We now illustrate the interest of multi-criteria BPT for multimodal imaging, in the context of radiological multi-image segmentation. It has to be noticed that the application of BPTs for medical image segmentation is innovative since BPTs have only been used in this context for anatomical object recognition [53].

1) Data: The proposed example involves both PET (Positron Emission Tomography) and standard CT (Computed Tomography) X-ray data. From these volumes, we extract 2D slices representing the lungs of the patient affected by a tumour lesion. The CT image provides homogeneous zones that characterise specific tissues and organs while the PET image provides local intensity minima where tumours are active, but with a spatial accuracy that is lower than CT information. Consequently, by coupling both grey-level value spaces into a single value space V, it may be possible to extract some homogeneous regions that gather the spatial accuracy of CT images and the spectral accuracy of PET ones, thus leading to accurate segmentation of organs and tumours. In the considered example (Figure 6(a,d)), the resolution of the registered images is 1318×864 and $V = [0, 255]^2$.

2) Method and results: To reduce the spatial complexity of this approach, the BPTs are built from an initial partition $\mathcal{L}$ composed of $862\,234$ regions obtained by applying a Mean-Shift clustering (spatial radius: 10, spectral value: 7, minimum region size: 12) on the CT image (Figure 6(a)). The valuation function $W_\star : (2^\Omega)^2 \times V^\Omega \to \mathbb{R}$ is defined as the increase of the ranges of the intensity values (for each image modality), potentially induced by the fusion of the incident regions.

The "standard" mono-image BPTs of the CT and PET images are first segmented independently by considering a user-defined horizontal cut, see Figure 6(b,e). The multi-image BPT, built from both the CT and the PET images, is then segmented in the same way, leading to the same number of regions, see Figure 6(c,f).

From the mono-image BPT results, one can note that different (but complementary) regions are obtained from the two modalities. The cut extracted from the mono-image BPT built from CT enables to capture different organs represented in the image (*e.g.*, rib cage, vertebral column) while the cut extracted from the mono-image BPT built from PET enables to delineate the main lesion from the lungs of the patient (see light-pink region in Figure 6(e)). The cut extracted from the multi-image BPT enables to directly gather in a same partition the organs extracted from the CT image and the lesion extracted from the PET image (Figure 6(c,f)). However, some homogeneous segments shown on Figure 6(b) are divided or modified on

Figure 6(c,f). We assume that the information carried by the PET image takes part in such phenomenon which may help to highlight unexpected sub-parts on the organs, that could correspond to lesions.

VII. Conclusion

In this article, we proposed a generalization of the BPT construction framework, classically built in a mono-feature way, thus allowing to consider various multi-feature paradigms. Such a multi-feature framework enables to reduce the task of the user, by offering more flexibility for the BPT creation. Indeed, the negotiation between the different features, at each step of the BPT construction, is intrinsically dealt with by the algorithm, with respect to the chosen consensus policies. This reduces the hard prior knowledge mandatory from the user to the only choice of the involved features and the global strategies for their collaboration. Experimental evaluations of this framework on two application cases highlight its versatility and its interest by demonstrating how it can be used to build consensual multi-feature BPTs from multiple images and / or multiple metrics computed through the image content.

The algorithmic evolutions related to the multi-feature BPT construction require the handling of more complex data-structures and consensual algorithms, compared to "standard" BPTs. In order to tackle the induced memory and time complexity issues raised by this generalized framework, the short-term perspective of this work will be to implement distributed heuristics based on graph-based distribution algorithmics. Integrating higher-level consensus may also allow us to improve the relevance of the hierarchies and the induced segmentation.

Beyond the application examples described in this article, other relevant applications could also be considered for the processing of different families of images. As an example, it is possible to apply multi-feature BPTs to segment hyperspectral images, by establishing a consensus between the complementary (and potentially correlated) information carried by the different spectral bands. Multi-temporal imaging can also be considered, by establishing a higher-level "temporal" consensus between the different image acquisitions of the same scene.

Another methodological challenge is raised by the possible divergences between the different values gathered by the metrics / images, which may lead to occasional irrelevant consensual decisions. We plan to study how non-consensual information could be used to follow local consensus between metrics / images leading to hypertrees where the branches model local fusion decisions.

References

[1] P. Salembier and M. H. F. Wilkinson, "Connected operators: A review of region-based morphological image processing techniques," *IEEE Signal Processing Magazine*, vol. 26, no. 6, pp. 136–157, 2009.

[2] L. Najman and J. Cousty, "A graph-based mathematical morphology reader," *Pattern Recogntion Letters*, vol. 47, pp. 3–17, 2014.

[3] "Mathematical morphology: From theory to applications," L. Najman and H. Talbot, Eds. ISTE/J. Wiley & Sons, 2010.

[4] P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval," *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 561–576, 2000.

[5] J. F. Randrianasoa, C. Kurtz, É. Desjardin, and N. Passat, "Multi-image segmentation: A collaborative approach based on binary partition trees," in *ISMM, International Symposium on Mathematical Morphology, Proceedings*, ser. Lecture Notes in Computer Science, vol. 9082. Springer, 2015, pp. 253–264.

[6] A. Rosenfeld, "Connectivity in digital pictures," *Journal of the ACM*, vol. 17, no. 1, pp. 146–160, 1970.

[7] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 641–647, 1994.

[8] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–598, 1991.

[9] S. L. Horowitz and T. Pavlidis, "Picture segmentation by a directed split-and-merge procedure," in *Second International Joint Conference on Pattern Recognition, Proceedings*, vol. 424, 1974, p. 433.

[10] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," *Computational Models of Visual Processing*, vol. 1, pp. 3–20, 1991.

[11] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[12] L. Grady, "Random walks for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.

[13] C. Couprie, L. Grady, L. Najman, and H. Talbot, "Power watershed: A unifying graph-based optimization framework," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 7, pp. 1384–1398, 2011.

[14] P. Salembier and J. Serra, "Flat zones filtering, connected operators, and filters by reconstruction," *IEEE Transactions on Image Processing*, vol. 4, no. 8, pp. 1153–1160, 1995.

[15] M.-M. Yau and S. N. Srihari, "A hierarchical data structure for multidimensional digital images," *Communications of the ACM*, vol. 26, no. 7, pp. 504–515, 1983.

[16] A. Montanvert, P. Meer, and A. Rosenfeld, "Hierarchical image analysis using irregular tessellations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 307–316, 1991.

[17] P. Salembier, A. Oliveras, and L. Garrido, "Anti-extensive connected operators for image and sequence processing," *IEEE Transactions on Image Processing*, vol. 7, no. 4, pp. 555–570, 1998.

[18] P. Monasse and F. Guichard, "Scale-space from a level lines tree," *Journal of Visual Communication and Image Representation*, vol. 11, no. 2, pp. 224–236, 2000.

[19] ——, "Fast computation of a contrast-invariant image representation," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 860–872, 2000.

[20] L. Najman and M. Schmitt, "Geodesic saliency of watershed contours and hierarchical segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1163–1173, 1996.

[21] B. Perret, S. Lefèvre, C. Collet, and E. Slezak, "Hyperconnections and hierarchical representations for grayscale and multiband image processing," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 14–27, 2012.

[22] R. Jones, "Connected filtering and segmentation using component trees," *Computer Vision and Image Understanding*, vol. 75, no. 3, pp. 215–228, 1999.

[23] L. Guigues, J.-P. Cocquerez, and H. Le Men, "Scale-sets image analysis," *International Journal of Computer Vision*, vol. 68, no. 3, pp. 289–317, 2006.

[24] N. Passat, B. Naegel, F. Rousseau, M. Koob, and J.-L. Dietemann, "Interactive segmentation based on component-trees," *Pattern Recognition*, vol. 44, no. 10–11, pp. 2539–2554, 2011.

[25] N. Passat and B. Naegel, "Component-trees and multivalued images: Structural properties," *Journal of Mathematical Imaging and Vision*, vol. 49, no. 1, pp. 37–50, 2014.

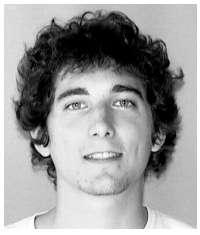
[26] C. Kurtz, B. Naegel, and N. Passat, "Connected filtering based on multivalued component-trees," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5152–5164, 2014.

[27] E. Carlinet and T. Géraud, "MToS: A tree of shapes for multivariate images," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5330–5342, 2015.

[28] N. Passat and B. Naegel, "Component-hypertrees for image segmentation," in *ISMM, International Symposium on Mathematical Morphology, Proceedings*, ser. Lecture Notes in Computer Science, vol. 6671. Springer, 2011, pp. 284–295.

[29] B. Perret, J. Cousty, O. Tankyevych, H. Talbot, and N. Passat, "Directed connected operators: Asymmetric hierarchies for image filtering and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 6, pp. 1162–1176, 2015.
[30] Y. Xu, T. Géraud, and L. Najman, "Connected filtering on tree-based shape-spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2015. [Online]. Available: DOI:10.1109/TPAMI.2015.2441070
[31] É. Grossiord, B. Naegel, H. Talbot, N. Passat, and L. Najman, "Shape-based analysis on component-graphs for multivalue image processing," in *ISMM, International Symposium on Mathematical Morphology, Proceedings*, ser. Lecture Notes in Computer Science, vol. 9082. Springer, 2015, pp. 446–457.
[32] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.
[33] P. Soille, "Constrained connectivity for hierarchical image partitioning and simplification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1132–1145, 2008.
[34] S. Valero, P. Salembier, and J. Chanussot, "Comparison of merging orders and pruning strategies for binary partition tree in hyperspectral data," in *ICIP, International Conference on Image Processing, Proceedings*, 2010, pp. 2565–2568.
[35] X. Giro and F. Marqués, "From partition trees to semantic trees," in *MRCS, Multimedia Content Representation, Classification and Security, Proceedings*, ser. Lecture Notes in Computer Science, vol. 4105. Springer, 2006, pp. 306–313.
[36] V. Vilaplana, F. Marques, and P. Salembier, "Binary partition trees for object detection," *IEEE Transactions on Image Processing*, vol. 17, no. 11, pp. 2201–2216, 2008.
[37] S. Valero, P. Salembier, and J. Chanussot, "Object recognition in hyperspectral images using binary partition tree representation," *Pattern Recognition Letters*, vol. 56, pp. 45–51, 2015.
[38] J. A. Benediktsson, L. Bruzzone, J. Chanussot, M. Dalla Mura, P. Salembier, and S. Valero, "Hierarchical analysis of remote sensing data: Morphological attribute profiles and binary partition trees," in *ISMM, International Symposium on Mathematical Morphology, Proceedings*, ser. Lecture Notes in Computer Science. Springer, 2011, pp. 306–319.
[39] C. Kurtz, N. Passat, P. Gançarski, and A. Puissant, "Extraction of complex patterns from multiresolution remote sensing images: A hierarchical top-down methodology," *Pattern Recognition*, vol. 45, no. 2, pp. 685–706, 2012.
[40] F. Calderero, F. Eugenio, J. Marcello, and F. Marques, "Multispectral cooperative partition sequence fusion for joint classification and hierarchical segmentation," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 6, pp. 1012–1016, 2012.
[41] C. Kurtz, A. Stumpf, J.-P. Malet, P. Gançarski, A. Puissant, and N. Passat, "Hierarchical extraction of landslides from multiresolution remotely sensed optical images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 87, pp. 122–136, 2014.
[42] S. Valero, P. Salembier, and J. Chanussot, "Hyperspectral image representation and processing with binary partition trees," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1430–1443, 2013.
[43] M. A. Veganzones, G. Tochon, M. Dalla Mura, A. J. Plaza, and J. Chanussot, "Hyperspectral image segmentation using a new spectral unmixing-based binary partition tree representation," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3574–3589, 2014.
[44] A. Alonso-González, C. López-Martínez, and P. Salembier, "Filtering and segmentation of polarimetric SAR data based on binary partition trees," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 2, pp. 593–605, 2012.
[45] P. Salembier, "Study of binary partition tree pruning techniques for polarimetric SAR images," in *ISMM, International Symposium on Mathematical Morphology, Proceedings*, ser. Lecture Notes in Computer Science, vol. 9082. Springer, 2015, pp. 51–62.
[46] A. Alonso-González, S. Valero, J. Chanussot, C. López-Martínez, and P. Salembier, "Processing multidimensional SAR and hyperspectral images with binary partition tree," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 723–747, 2013.
[47] A. Alonso-González, C. López-Martínez, and P. Salembier, "PolSAR time series processing with binary partition trees," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 6, pp. 3553–3567, 2014.
[48] L. Garrido, P. Salembier, and D. Garcia, "Extensive operators in partition lattices for image sequence analysis," *Signal Processing*, vol. 66, no. 2, pp. 157–180, 1998.
[49] C. Kurtz, N. Passat, A. Puissant, and P. Gançarski, "Hierarchical segmentation of multiresolution remote sensing images," in *ISMM, International Symposium on Mathematical Morphology, Proceedings*, ser. Lecture Notes in Computer Science, vol. 6671. Springer, 2011, pp. 343–354.
[50] J. Cousty, L. Najman, and B. Perret, "Constructive links between some morphological hierarchies on edge-weighted graphs," in *ISMM, International Symposium on Mathematical Morphology, Proceedings*, ser. Lecture Notes in Computer Science, vol. 7883. Springer, 2013, pp. 86–97.
[51] V. Machairas, M. Faessel, D. Cárdenas-Peña, T. Chabardes, T. Walter, and E. Decencière, "Waterpixels," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3707–3716, 2015.
[52] A. Al-Dujaili, F. Merciol, and S. Lefèvre, "GraphBPT: An efficient hierarchical data structure for image representation and probabilistic inference," in *ISMM, International Symposium on Mathematical Morphology, Proceedings*, ser. Lecture Notes in Computer Science, vol. 9082. Springer, 2015, pp. 301–312.
[53] T. Blaffert, "Recognition of anatomically relevant objects with binary partition trees," in *ICIP, International Conference on Image Processing, Proceedings*, vol. 3, 2001, pp. 34–37.

Jimmy Francky Randrianasoa obtained the MSc degree in Computer Science and ESIROI Engineer degree, from the Université de la Réunion, France, and the Engineer degree, from ENI, Madagascar, in 2014. He is currently working toward the PhD degree at Université de Reims Champagne-Ardenne, France. His research interests include image analysis, mathematical morphology and remote sensing.



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

Éric Desjardin obtained the PhD degree in Computer Science from Université de Reims Champagne-Ardenne, France, in 1993. He is now an associate professor in the same university. He worked on character recognition and ancient text understanding until 2000. His current interest focuses on AI approaches to geographical information system.

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