Binary Partition Tree Construction from Multiple Features for Image Segmentation

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

Binary Partition Tree Construction from Multiple Features for Image Segmentation[☆]

Jimmy Francky Randrianasoa[a,*], Camille Kurtz[b], Éric Desjardin[a], Nicolas Passat[a]

[a] *Université de Reims Champagne-Ardenne, CReSTIC, France*
[b] *Université Paris Descartes (Sorbonne Paris Cité), LIPADE, France*

Abstract

In the context of digital image processing and analysis, the Binary Partition Tree (BPT) is a classical data-structure for the hierarchical modelling 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 embed multiple features. 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. In particular, this provides alternatives to the complex issue of arbitrary metric construction from several – possibly non-comparable – features. 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 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 metrics computed through the (potentially multiple) image content(s), in particular in the context of remote sensing.

Keywords: binary partition tree, morphological hierarchy, multiple features, graph-based image processing, image segmentation

1. Introduction

1.1. 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 modelling 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 modelled as a tree structure, whose each node is a connected region. Each of these nodes is either a leaf –an elementary region– or an internal node, modelling 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.

[☆]This work was supported by the French *Agence Nationale de la Recherche* under Grant ANR-12-MONU-0001.

[*]Corresponding author
Email address: jimmy.randrianasoa@univ-reims.fr (Jimmy Francky Randrianasoa)

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.

1.2. Motivations and Contributions

The BPT has already demonstrated its relevance for challenging image processing and analysis tasks, for instance in the fields of video analysis, 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, the metric –namely the merging criterion used to decide, at each step of the BPT construction, which nodes to merge– is a scalar function, that imposes to fuse various (potentially non-comparable) elements of expert knowledge (e.g., colorimetric and geometric heterogeneity criteria); 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. On the other 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.

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 metrics and / or many images*. 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 3.

In a second time, we explain in Section 4, how the basic BPT construction framework can be generalized to handle multiple features. We first identify the requirements in terms of data-structures (Section 4.1), and then the algorithmic side effects (Section 4.2). A complexity analysis (Section 4.3) is then proposed to describe the induced space and time complexity increases.

Based on this theoretical framework, we present in Section 5 some practical implementation details of the proposed tool. For the sake of reproducibility, we provide an open-source library (Section 5.1) for the creation of multi-feature BPTs, which also constitutes a technological contribution of this work. We then detail the proposed algorithms (Section 5.2) and the main data-structures (Section 5.3) implemented in our library to construct multi-feature BPTs.

In a fourth time, we discuss in Section 6 some perspective strategies for scaling up the multi-feature construction. To this end, we identify the main bottlenecks when considering standard sequential algorithmics (Section 6.1). We then consider two points of view: first, heuristic solutions that approximate the exact algorithm in a sequential way (Section 6.2); second, the way to switch from a sequential to a distributed algorithm (Section 6.3).

This work is concluded by two application examples in the domain of satellite image analysis, in Section 7, that illustrate how the framework can allow us to define BPTs from multiple metrics and from multiple images.

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 2; and a conclusion that emphasises the perspectives of this work, in Section 8.

2. Related works

2.1. 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 modelled by graph edges. 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], hyper-connected component-trees [21], and binary partition trees [4] (see Section 2.2). 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), further developments were proposed to allow for a better flexibility in image and parameter handling. The case of multiband (e.g., 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].

2.2. 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 multi-temporal SAR image analysis [47].

2.3. 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 colour 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 the richness of the information carried by the diversity 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 for the potential optimization of our algorithmic framework, and its distribution over several processing cores.

3. Structural Description of the BPT Construction

3.1. 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 define an adjacency inherited from that of Ω. 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.

3.2. 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 node modelling the union of two neighbouring regions. The root node corresponds to the image support.

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 modelled 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 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., colour, 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 colour in a specific 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 as region model 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.

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

3.3.1. 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 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.

3.3.2. 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 assumed to be connected with respect

[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.

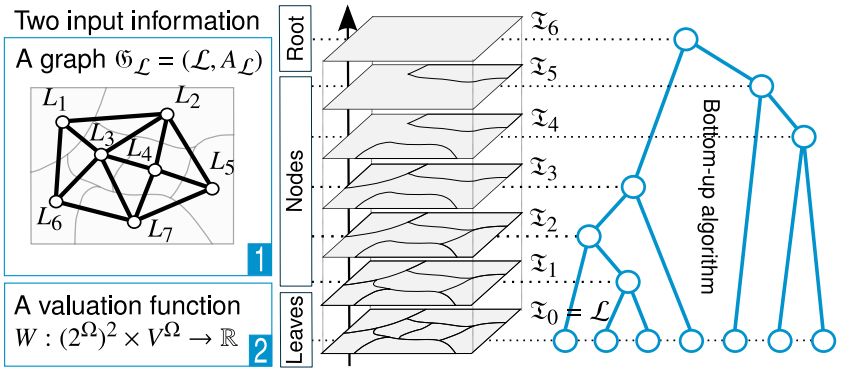


Figure 1: Illustration of the algorithm for the creation of a BPT from one image. Left: two input information which are (1) a graph $\mathfrak{G}_\mathcal{L} = (\mathcal{L}, A_\mathcal{L})$ that models the initial partition of the image; and (2) a valuation function $W : (2^\Omega)^2 \times V^\Omega \to \mathbb{R}$ that allows us to choose, at each step of the process, the next pair of nodes to be merged (see Section 3.3). Right: progressive, bottom-up, creation of the tree from $\mathfrak{T}_o$ to $\mathfrak{T}_6$ by iterative fusions of two neighbouring regions.

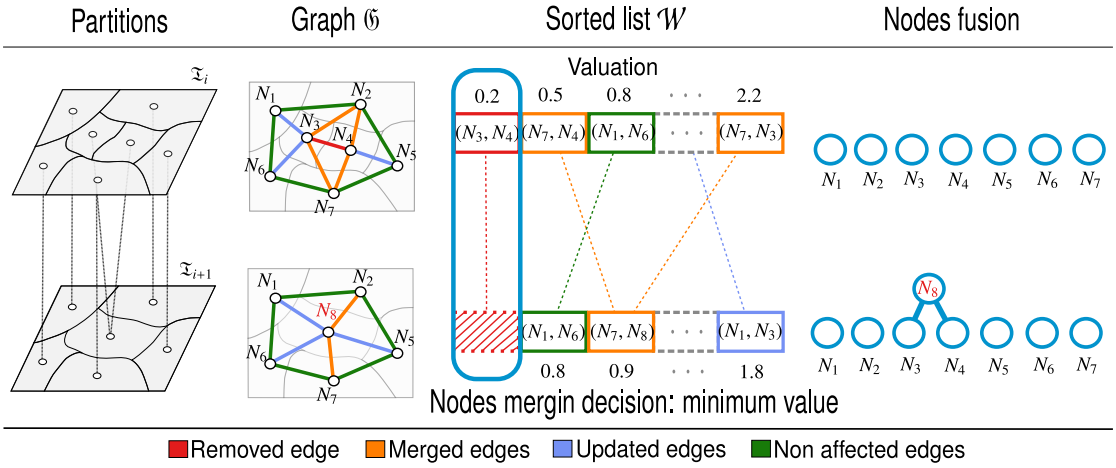


Figure 2: One step of the building of a BPT from one image. From the left to the right: First: the partition of Ω before and after the fusion of two nodes. Second: the associated graph $\mathfrak{G}$, before and after the fusion of N_3 and N_4, forming the new node N_8; the red edge is removed; the blue 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.

to A_Ω.) This partition $\mathcal{L}$ defines the set of the BPT leaves 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 $\mathfrak{G}_n$ ($\mathfrak{T}$ 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

[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 *a priori* on the initial partition.

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 $\mathfrak{T}_n$.

3.3.3. 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 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 $\mathcal{O}(\alpha.\log_2|\mathcal{W}|)$.

4. Multi-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 use *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 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.

4.1. Structural Evolutions

The proposed generalization deals with the "feature" part of the construction. As stated in Section 3.3.1, 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 modelled 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 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 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 modelled by a sorted list $\mathcal{W}$ of the values of all graph edges. This list was updated during the progressive collapsing of $\mathfrak{G}$, by removing elements from the list; updating the values of some edges (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. By assuming that we consider k distinct images, and l distinct metrics, we may 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 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 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 algorithm described in Section 3.2, which initially depended on one list $\mathcal{W}$.

4.2. Algorithmic Evolutions

From an algorithmic point of view, each iteration of the construction process preserves the same organization. An

[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.).

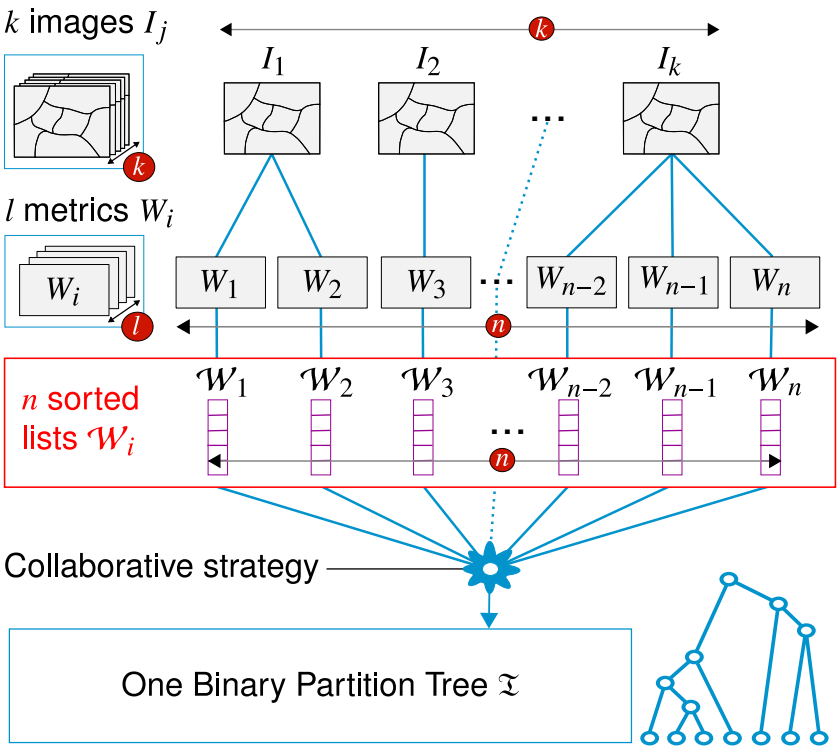


Figure 3: General structure of a BPT creation involving n valuation functions / sorted lists. Each list $\mathcal{W}_i$ gathers edges information computed by using one valuation function corresponding to a given metric on an image. A BPT is obtained by applying a collaborative strategy leading to a consensus between the information carried by these n lists (see Section 4.2).

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.

4.2.1. 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.

4.2.2. 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 lists $\mathcal{W}_i$. These consensus (namely, *majority vote* and *most frequent*, potentially

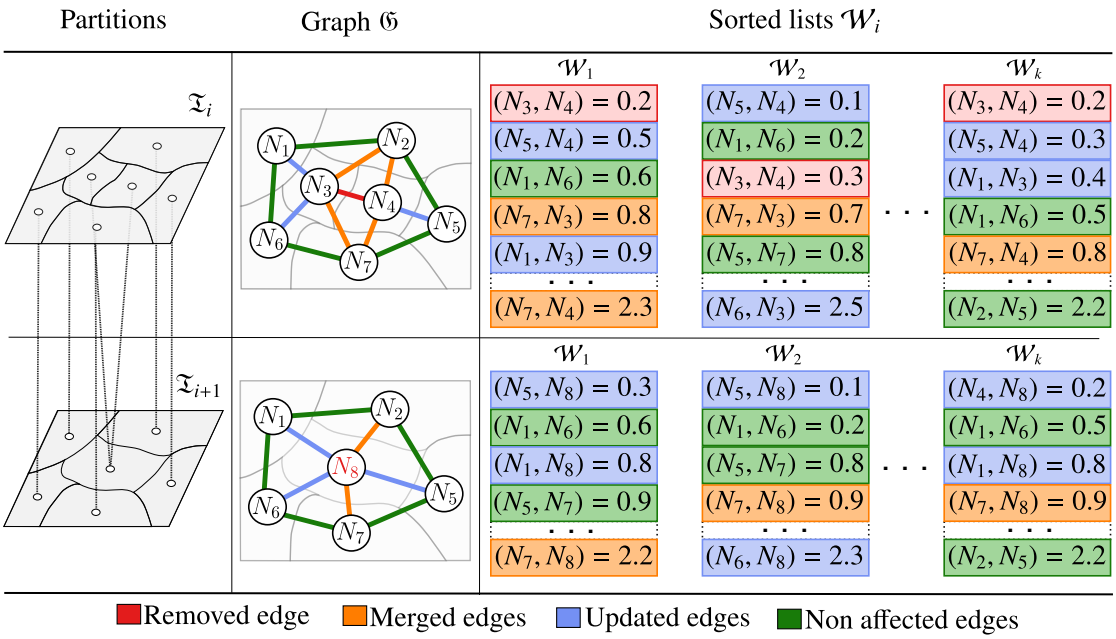


Figure 4: One step of the building of a BPT involving n lists $\mathcal{W}_i$. Left: the partition of Ω before and after the fusion of two nodes. Center: the associated graph $\mathfrak{G}$, before and after the fusion of N_3 and N_4, forming the new node N_8; the red edge is removed; the blue and orange 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.

weighted) policies do not act on the absolute valuations of the edges, but on their relative positions in the lists. Another strategy can also consist in choosing the edge with the lowest *mean of ranks*, according to the position of the edges within the lists. In such cases, it is then mandatory to maintain n sorted lists. However, the decision process does not require to 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.

4.2.3. 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 need not 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.

4.3. 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 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 1.

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 4.2.1). Following the same kind of calculation, the cost for the second consensus policy (Section 4.2.2) is $\mathcal{O}(n.|\mathcal{L}|\log_2|\mathcal{L}|)$, while that of the third (Section 4.2.3) is $\mathcal{O}(n.|\mathcal{L}|^2)$.

5. Implementation details

For the sake of reproducibility, we provide in this section some implementation details of the proposed algorithms and data-structures that are required to compute multi-feature BPTs.

5.1. AGAT*: An open-source library for multi-feature BPT construction*

As a technological contribution of this work, we developed an open-source JAVA library[4], called AGAT, that implements the proposed algorithm and the required data-structures for the creation of a multi-feature BPT (called MBPT in the following). It should be noticed that other open-source libraries were already proposed to create BPTs [47, 51]; however these former libraries rely on the creation of classical mono-feature BPTs.

The AGAT library is generic, since it gives the user the opportunity to totally tune the construction of their MBPT, by easily implementing their own metric and consensus strategies as new JAVA classes. To this end, our library uses the same pattern factory mechanism for both cases, and the user only needs to create new classes that extend either a predefined metric or a consensus class parent. To create a new metric, two methods have to be designed: the first defining the computation of the underlying feature value for a particular node, and the second defining the metric (dissimilarity) computation between two adjacent nodes according to this feature. To create a new consensus, the user needs to implement a method that applies the desired consensus strategy and returns the next couple of nodes to be merged (see Section 5.2.2).

We have also developed and integrated in AGAT a TIFF library which allows the user 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.

Once a MBPT has been constructed, a saving system implemented in the library enables to store the tree structure in a file. Such file, encoded in the JSON format, contains information about the tree, such as the pixels contained in each leaf, the merging orders of the nodes, the paths of the images in the file system, the metrics and the consensus strategy used. Thanks to this JSON file, the MBPT can be rebuilt as many times as wanted, without recomputing it. Indeed, the library can emulate the construction of the tree (defining leaves, merging nodes and getting the root) by simply reading the JSON file.

For image segmentation purpose, a standard cut can be performed from the MBPT file, by specifying the number of desired regions. As output of this cutting method, the framework generates segmentation images such as border image and label images based on random colours. Supplementary image results can be generated if the user wants to produce, for example, the segmentation results related to a specific interval of region numbers. Many other segmentation paradigms based on BPTs have been proposed in the literature, mainly based on energy optimization, or criterion-based node selection. These paradigms are not natively implemented in the proposed library, whose principal purpose is the construction of MBPTs; however, they can be easily interfaced with AGAT.

5.2. Implementation details of the proposed algorithm

To build a multi-feature BPT, the main structure of the classical BPT creation algorithm remains unchanged. Indeed, the MBPT is still 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. However, instead of dealing with a classical merging criterion only relying on one image and one metric, our algorithm may now consider many images, different metrics, and a consensus strategy as input parameters. The core of the algorithm is divided as:

1. preparation of the lists $\mathcal{W}_i$ based on each image and metric couples;
2. preparation of the graph $\mathfrak{G}$ modelled as a Region Adjacency Graph (RAG);
3. creation of the MBPT tree $\mathfrak{T}$ by merging pairs of nodes.

The general structure of the proposed algorithm is presented in Algorithm 1.

5.2.1. Preparation of the lists $\mathcal{W}_i$ and the RAG

As already mentioned in Section 4.1, by assuming that we consider k distinct images and l distinct metrics, we may have up to $n = k.l$ valuation functions. Each image is then associated to a specific metric. This leads us to prepare no longer one, but n sorted lists $\mathcal{W}_i$. Each list $\mathcal{W}_i$ will contain all edges of the graph $\mathfrak{G}$ and will be maintained ordered during the creation process of the MBPT. Its data-structure and maintenance will be detailed in Section 5.3.5. An object representing the metric is instantiated by the algorithm and, linked to each list $\mathcal{W}_i$. At this stage of the process, those n lists are still empty but ready.

When all lists $\mathcal{W}_i$ are prepared and linked to the right couple of image and metric, a graph $\mathfrak{G}$ (implemented as

[4] Available at `https://bitbucket.org/agat-team/agat-v0.3`.

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

Table 1: Cost of the BPT construction for various families of consensus. For readability purpose, α and $T_{W_\star}$, which are practically bounded by low constant values have been omitted.

Algorithm 1: General structure of the proposed algorithm

Data: Image $I[\,]$; Metric $W[\,]$; Consensus strategy *Consensus*;
Result: One Multi-feature Binary Partition Tree $MBPT$;

```
/* Variables                                                    */
Region Adjacency Graph 𝔊;                                  // the RAG
List 𝒲[ ];                                  // the list of lists 𝒲_i
Image preSeg;          // an initial partition of the image support
/* Initializations                                              */
prepareLists(I, W);            // associate metrics, images & adjacency lists
prepareRAG(𝒲, preSeg);         // build initial leaves & adjacencies
fillLists(𝒲, 𝔊);               // fill the lists with the initial adjacencies
/* Node fusions while the list of adjacency is not empty       */
while 𝔊.tabAdja is not empty do
    Adjacency chosenAdja = Consensus.apply(𝒲);
    Node n1 = chosenAdja.region1;
    Node n2 = chosenAdja.region2;
    Node newNode = fusionOf(n1, n2);
    𝔊.tabNode.add(newNode);
    𝔊.tabAdja.update();    // adding, removing and updating values
    /* Update the content of the lists                          */
    for i = 0; i < 𝒲.size(); i++ do
        𝒲[i].update();     // adding, removing and updating values
MBPT.tabNode = 𝔊.tabNode;
return MBPT
```

a RAG) has to be built. By starting the process from the pixels or from an initial partition of the image support (e.g., a pre-segmented image: superpixels, flat zones, etc.), the initial nodes of the graph are created. These nodes also constitute the set of leaves of the MBPT. Each leaf represents an initial sub-part of the image(s) and contains information about the corresponding pixels. Once the leaves are prepared, the edges of the graph are created and their values are computed by invoking particular methods of the corresponding metric. Those edges represent the adjacencies between neighbouring regions of the image. For each edge, n computed metric values are associated. In parallel, we use these metric values to fill the n lists $\mathcal{W}_i$ that should be further maintained sorted.

5.2.2. Creation of the MBPT by merging nodes

Once the lists $\mathcal{W}_i$ and the graph $\mathfrak{G}$ are ready, an iterative fusion of the nodes can be operated to build the MBPT in a bottom-up fashion. This iteration process stops when the lists $\mathcal{W}_i$ are empty. This state means that the root of the tree is reached. The main part of this iterative block is composed of:

1. the choice of the two next adjacent nodes of $\mathfrak{G}$ to be merged by applying the consensus strategy defined by the user;
2. the handling of the n lists $\mathcal{W}_i$ by removing, in each list, the adjacency linking the two chosen nodes and by updating the values of all adjacencies that link them to their neighbours;
3. the resorting of the lists.

5.3. Implementation details of the data-structures

Let us now describe the main data-structures used by the algorithm and implemented in our library. For a better comprehension of their implementation, only the most important structures such as the MBPT, the RAG, the nodes, the adjacencies and the lists $\mathcal{W}_i$ are mentioned here. Note that some of these data-structures are strongly dependent on the Java philosophy.

5.3.1. Data-structure of the MBPT

In the proposed library, the MBPT is modelled by a class whose most important attributes are: a list of the image(s); a list of lists $\mathcal{W}_i$; the RAG; the consensus strategy to be used; and an indexation matrix to map the pixels from the image support to the nodes. Another optional attribute of this class is a pre-segmented image (i.e., an initial partition of the image support) if the user does not want to start the creation of the MBPT from the pixels.

This MBPT class does not actually represent a physical structure of tree; however it gathers all the required information for its creation and management. Its most important methods are: the growing method, which is the principal element of the creation process, since it contains the three main points of the algorithm described in Section 5.2; the regrowing method that allows for the rebuilt of the MBPT from a saved Json structured file; and finally the cutting method that generates segmentation results.

5.3.2. Data-structure of the RAG

As an attribute of the MBPT class, the RAG contains simultaneously the structure of the tree $\mathfrak{T}$, and the structure of the graph $\mathfrak{G}$. In particular, it contains a table of all nodes that regroups the nodes of both the graph $\mathfrak{G}$ and the tree $\mathfrak{T}$. Indeed, each node of the graph is also one of the tree. This table of nodes is static and the required memory space allocation is only done once when preparing the RAG.

It also contains a static table of all adjacencies, so its memory allocation space needs not be done more than once. This table regroups all the active adjacencies of the graph $\mathfrak{G}$. Unlike the lists $\mathcal{W}_i$, this set is not ordered. Its main purpose is to store the adjacencies in the memory. During the creation process of the tree, the content of this table can only decrease. Indeed, some adjacencies are deleted and others are updated during the node merging iterations. When an element is removed from this table, it is replaced by the last not empty element. By doing so, all empty cells remain at the end of the table, leading to a compact structure (making easier iteration in its content when needed). At the end of the node merging iterations, this table is totally empty.

5.3.3. Data-structure of a node

The nodes represent important elements of both the MBPT tree $\mathfrak{T}$ and the graph $\mathfrak{G}$. Each node is modelled by a class whose principal attributes are: the type of the node (leaf, simple node or root); a list of pixels contained in the region modelled by the node; the left and right children of the node (and its father); a list of adjacencies linking the node to its neighbours; and a list of feature values associated to all metrics and prepared by each of them.

5.3.4. Data-structure of an adjacency

The adjacencies are edges representing the links between each neighbouring regions. Each of them are associated to n values computed from each metric. Each adjacency is modelled by a class and its attributes are:

- a couple of neighbouring nodes;
- a table of metric values representing the distances between the two nodes (relatively to the n metrics);
- an index giving its position in the table of adjacencies of the RAG;
- a table of ranks;
- two tables of previous and next adjacencies.

The three last attributes are used during the handling of the n lists $\mathcal{W}_i$ by particular consensus strategies. Their interest will be explained in the next section. Each adjacency also implements some methods of the Java comparable interface. Surcharging those methods allows for the comparison between adjacencies, that helps their sorting in the lists $\mathcal{W}_i$ by using the table of metric values.

5.3.5. Data-structure of a list $\mathcal{W}_i$

The data-structures used for the n lists $\mathcal{W}_i$ are crucial in this algorithm. Indeed, most instructions related to the MBPT creation rely on them. Each list $\mathcal{W}_i$ contains all adjacencies of the RAG (see Section 5.3.2) and must be maintained sorted. The sorting process of each list is realized by considering the corresponding metric values (also called scores) saved in each adjacency (see Section 5.3.4). The data-structure of the lists $\mathcal{W}_i$ must be designed carefully in order to lighten the MBPT creation process. In Agat, a list $\mathcal{W}_i$ is modelled by a class containing an hybrid data-structure based on a red-black binary search tree, providing guaranteed logarithmic time cost for basic operations. An illustration of this hybrid data-structure is provided in figure* 5, where each box represents an adjacency and the bold relations model father-son relationships in the binary search tree. This structure is maintained sorted thanks to a specific comparator that determines which adjacency has to be placed before or after another. In our case, if some adjacencies have the same score s, the least recent are placed before the most recent (e.g., Adja. 6 and Adja. 4 in figure* 5).

The proposed structure of a list $\mathcal{W}_i$ is particular because its implementation allows also each adjacency to know its rank r in each list. Then, the process can get rapidly the position of the adjacency in a list $\mathcal{W}_i$. Such strategy is useful to speed up the computation carried out by the consensus strategies, in particular in the case of relative and global consensus strategies (see Sections 4.2.2 and 4.2.3): for instance, to choose the adjacency with the lowest mean of ranks, according to its positions within the n lists $\mathcal{W}_i$. Indeed, the ranks of each adjacency can be determined rapidly without having to iterate on all lists content. When a list $\mathcal{W}_i$ is modified (adding, removing, sorting elements), three specific attributes of each adjacency (see Section 5.3.4) are updated:

- the table of ranks;
- the table of previous adjacencies;
- the table of next adjacencies.

Those attributes allow for the formation of a linked list of adjacencies (combined to the red-black binary search tree), whose links are built according to the ranks of the adjacency in the list $\mathcal{W}_i$ (white boxes model pointers while dashed relations model linkages in figure* 5). In order to reduce the task and time computation related to the modification of each list, the ranks are only updated once per node fusion.

5.4. Parallel streaming on the list of lists $\mathcal{W}_i$

The Java 8 proposed an API stream which simplifies the processing of the collections, the tables and the I/O sources. The stream is not based on the iteration pattern and does not store additional data, but only manages to queue them for processing. This allows us to reduce the

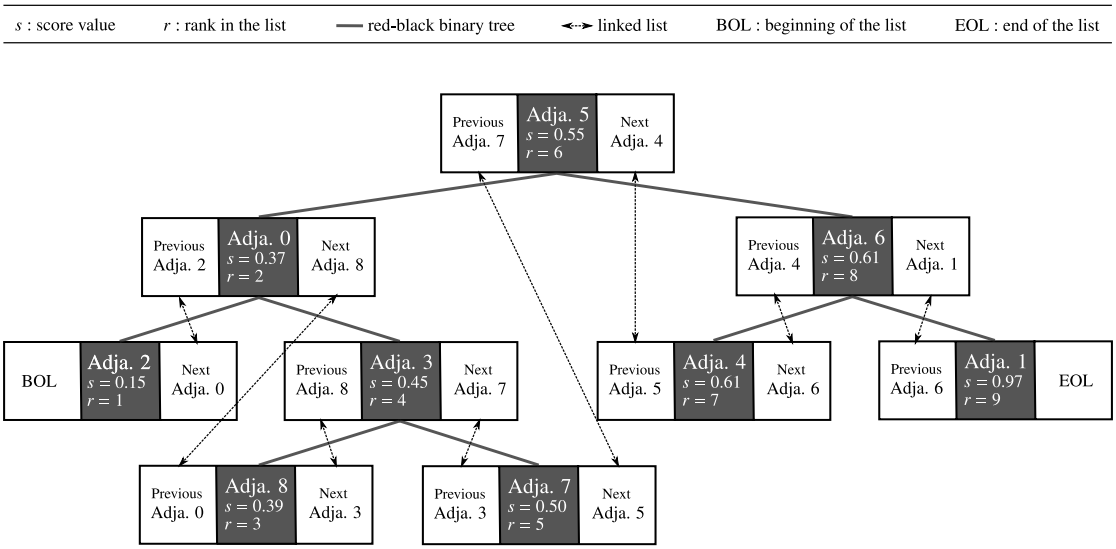


Figure 5: Illustration of the hybrid data-structure of a list $\mathcal{W}_i$ containing all adjacencies of the RAG (9 in this example) represented as black boxes. The list is implemented using a red-black binary search tree, maintained sorted according to the metric values (also called scores s) saved in each adjacency. The bold relations model the father-son relationships of the binary search tree. On top of this structure, a linked list of adjacencies is implemented (see Section 5.3.5) to store the ranks r of each adjacency in the list $\mathcal{W}_i$: white boxes model pointers (previous and next elements in the list) while dashed relations model double linkages.

memory consumption resulting from the creation of temporary variables during the iteration of a list content. In addition, the reason why we used the streaming concept on the list of lists $\mathcal{W}_i$ (see Section 5.3.1) is justified by its ability to process them in a parallel and independent way. Indeed, the streaming system relies on a fork / join framework provided by the Java SE. Actually, when the n lists $\mathcal{W}_i$ need to be modified or updated, the stream object puts each list $\mathcal{W}_i$ in a sub-stream and threads them for a parallel processing.

6. Perspectives for scaling up the Multi-feature BPT Construction

6.1. Bottlenecks

The complexity analysis presented in Section 4.3 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 bottlenecks, thus leading to algorithms that will lead to fairly similar results as the exhaustive algorithm, but with a lower complexity.

6.2. 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 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 [52] 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 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 [51]. 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 sizes, 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.

6.3. 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 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.

6.3.1. 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 linear, and we preserve the same cost as for standard BPT construction for these parts of the 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.

6.3.2. 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.

7. Experimental studies

To illustrate our framework, two application cases have been considered in the domain of remote sensing image analysis. 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. In Section 7.1, we show how a multi-feature BPT can be built from a single complex satellite image by considering simultaneously, and in a consensual manner, various metrics. In Section 7.2, we show how a multi-feature BPT can be built from multiple satellite images sensed over the same geographical area by considering simultaneously information provided by different image contents.

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 complex imaging domains.

7.1. 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 images.

7.1.1. Data

The dataset used here (courtesy LIVE, UMR CNRS 7263) was sensed over the town of Strasbourg (France) by the PLÉIADES satellite in 2012. The first sample is a VHSR image (2000×2000 pixels) representing a complex high-density urban area (figure* 6(a)) composed of different urban objects (e.g., individual houses, industrial buildings, parking lots, roads, shadows, water canals). The second sample is also a VHSR image (2000×2000 pixels) but that represents a typical low-density urban area (figure* 6(d)) composed of different geographical objects (e.g., crop fields, forests, bare soils, rivers). These two satellite images are pansharpened multispectral images at a spatial resolution of 60 cm with four spectral bands (R, G, B, NIR).

7.1.2. Method and results

To reduce the spatial complexity of this approach, the BPTs are built from an initial partition $\mathcal{L}$ composed of 200000 regions obtained from a cut performed on a "standard" BPT.

We considered here four complementary valuation functions $W_\star : (2^\Omega)^2 \times V^\Omega \to \mathbb{R}$ that model radiometrical or geometrical information related to region dissimilarities.

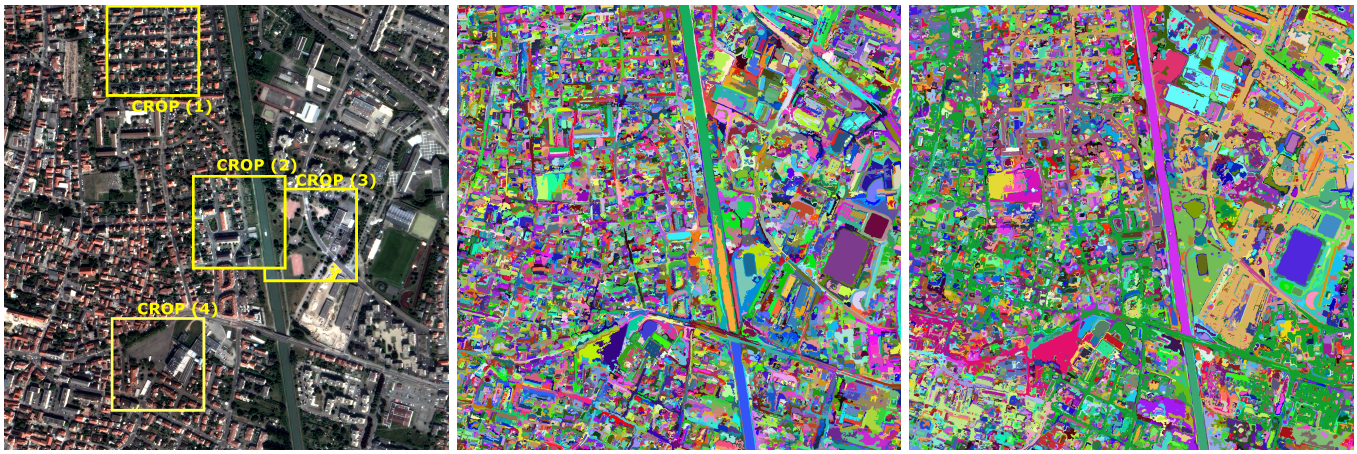
The first valuation function W_{colour} is defined as the increase of the ranges of the pixel intensity values for each radiometric band, potentially induced by the fusion of incident regions (i.e., a virtual merge). Practically, W_{colour} can be computed as follows. Let $N_i, N_j \in \mathcal{P}$ be two distinct and adjacent regions / nodes. A multispectral image is modelled as a function $\mathcal{I} : \Omega \to V$ which associates, to each point $x \in \Omega$, a s-uple (with $s > 1$ the number of spectral bands) of spectral intensities $\mathcal{I}(x) = \prod_{b=1}^{s} \mathcal{I}_b(x)$. The W_{colour} valuation function is then computed as $\frac{1}{s}\sum_{b=1}^{s} \max\{v_b^+(N_i), v_b^+(N_j)\} - \min\{v_b^-(N_i), v_b^-(N_j)\}$ where $v_b^\star$ provides the extremal values for the b-th spectral band in $\mathcal{I}$ (*i.e.*, in $\mathcal{I}_b$).

The second valuation function W_{ndvi} quantifies the difference of NDVI between two adjacent regions. The NDVI (Normalized Difference Vegetation Index) of a pixel is a simple indicator that can be used to analyze remote sensing data and assess whether the target being observed contains live green vegetation or not. It is simply computed as a function that associates to each point $x \in \Omega$ the ratio $(I_{NIR}(x) - I_R(x))/(I_{NIR}(x) + I_R(x))$ involving both the R and NIR channels. Practically, the W_{ndvi} valuation function is computed for two adjacent nodes N_i, N_j as the absolute difference between the averages of NDVI associated to the pixels of the two regions.

The third valuation function W_{ndwi} computes the difference of NDWI (Normalized Difference Water Index) between two adjacent regions. The NDWI of a pixel is an indicator to assess whether the target being observed contains presence of water or not. The W_{ndwi} valuation function is computed similarly to the W_{ndvi} valuation function by replacing the R channel by the G one.

The fourth valuation function W_{elong} is defined as the change of the geometrical elongation values, potentially induced by the fusion of two incident regions. The geometrical elongation of a node N is computed as the ratio of the height and width of its bounding box and is denoted as $elong(N)$. Practically the W_{elong} valuation function is computed for two adjacent nodes N_i, N_j as $|elong(N_i \cup N_j) - \frac{elong(N_i)+elong(N_j)}{2}|$.

In the case of multi-criteria BPTs, the relative local



(a) High-density urban area. (b) BPT of (a). (c) MBPT of (a).

(d) Low-density urban area. (e) BPT of (d). (f) MBPT of (d).

Figure 6: (a, d) Illustration of two very-high spatial resolution satellite images (2000×2000 pixels) at a spatial resolution of 60 cm sensed by the PLÉIADES satellite and covering different geographical areas. (b, resp. e) Segmentation result from a cut containing 23500 regions (resp. 5000 regions) performed on a "standard" BPT of (a, resp. d). (c, resp. f) Segmentation result from a cut containing 23500 regions (resp. 5000 regions) performed on a MBPT of (a, resp. d) using four features: the colour intensity, the region elongation, NDVI and NDWI values $\{W_{colour}, W_{elong}, W_{ndvi}, W_{ndwi}\}$.

information consensus policy *mean-of-ranks*, according to the position of the edges within the lists is applied for the first 15% of the lists $\mathcal{W}_\star$.

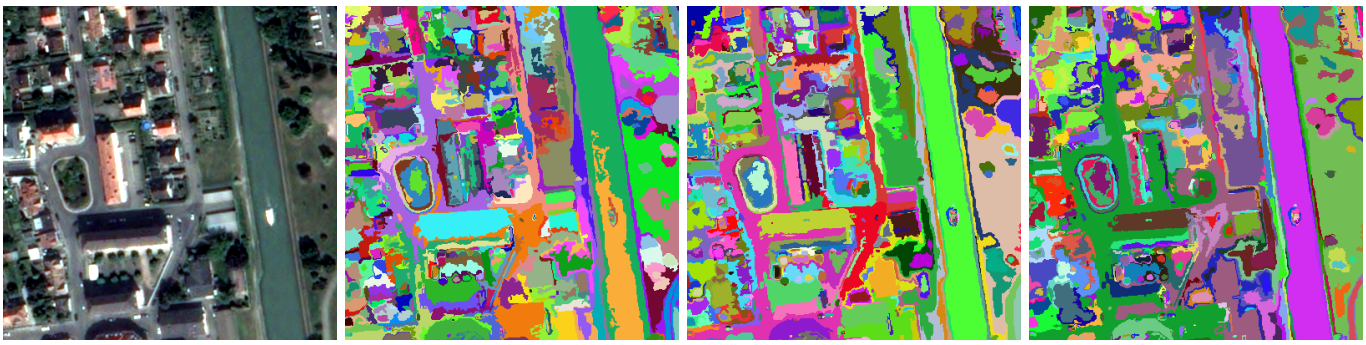
As a baseline, the "standard" BPTs of the satellite images of figure* 6(a, d) are constructed using the colour intensity value criterion W_{colour}. These trees are then segmented by considering a user-defined horizontal cut to produce a partition whose the region scales are adapted to segment the various objects contained in the sensed geographical areas (e.g., individual houses, buildings, roads, forests), see figure* 6(b, e). For visualisation purpose, the segmentation results are depicted here in random false colours.

In order to evaluate the impact of the different valuation functions $W_\star$ on the segmentation results, the multi-criteria BPTs are then built by considering various combinations of valuation functions (e.g., $\{W_{colour}, W_{ndvi}\}$, $\{W_{colour}, W_{elong}, W_{ndvi}\}$, $\{W_{colour}, W_{elong}, W_{ndvi}, W_{ndwi}\}$). The produced multi-criteria trees are then segmented in the same way as for the "standard" BPTs, leading to the same number of regions. For illustration purpose on the entire images, figure* 6(c, f) shows some examples of the best results obtained from our multi-criteria BPTs.

From the "standard" BPT results (see figure* 6(b, e)), we observe that the obtained regions are quite radiometrically homogeneous and are adapted to extract simple urban objects (e.g., small road segments, bare soils). Concerning more complex objects strongly structured by their geometrical shapes (e.g., house roofs, rivers, large vegetation areas and roads with elongated structures), the regions produced with the "standard" BPT are not always relevant since the considered urban objects are often composed of several small sub-regions (see the river, road sections and vegetation areas in figure* 6(b, e)). In comparison, the cut extracted from the MBPT enables to directly

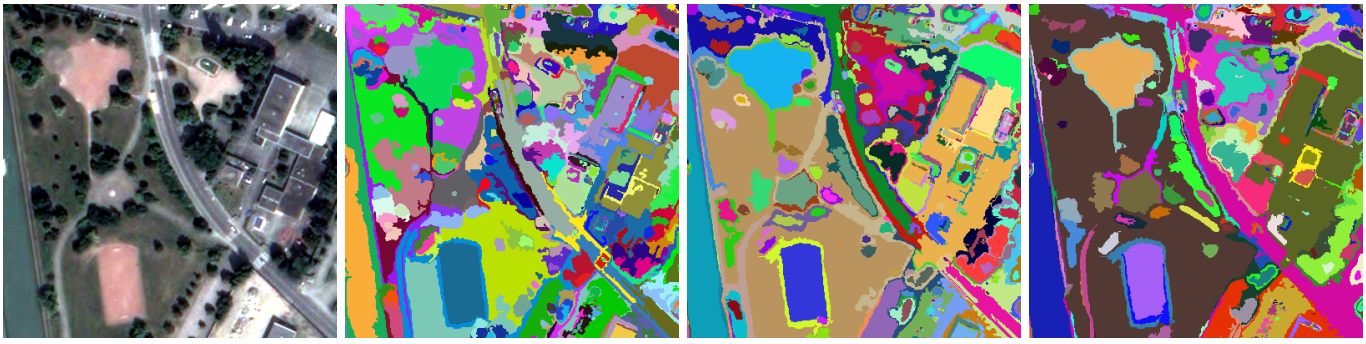
(a) Image crop (1). (b) BPT. (c) MBPT-1. (d) MBPT-2.

Figure 7: Segmentation results from the BPTs and the MBPTs, centered on crop (1) of the image presented in figure* 6(a). BPT: $\{W_{colour}\}$; MBPT-1: $\{W_{colour}, W_{ndvi}\}$; MBPT-2: $\{W_{colour}, W_{elong}, W_{ndvi}, W_{ndwi}\}$.



(a) Image crop (2). (b) BPT. (c) MBPT-1. (d) MBPT-2.

Figure 8: Segmentation results from the BPTs and the MBPTs, centered on crop (2) of the image presented in figure* 6(a). BPT: $\{W_{colour}\}$; MBPT-1: $\{W_{colour}, W_{ndwi}\}$; MBPT-2: $\{W_{colour}, W_{elong}, W_{ndvi}, W_{ndwi}\}$.



(a) Image crop (3). (b) BPT. (c) MBPT-1. (d) MBPT-2.

Figure 9: Segmentation results from the BPTs and the MBPTs, centred on crop (3) of the image presented in figure* 6(a). BPT: $\{W_{colour}\}$; MBPT-1: $\{W_{colour}, W_{ndvi}, W_{ndwi}\}$; MBPT-2: $\{W_{colour}, W_{elong}, W_{ndvi}\}$.

gather, in a same partition, regions corresponding both to simple urban objects and to complex ones (see the previous mentioned geographical objects in figure* 6(c, f)).

To facilitate the qualitative analysis of the obtained results, we focus hereinafter on eight illustrative image crops (from figure* 7 to figure* 14) extracted from the two satellite images of the dataset. Note that the BPTs and MBPTs are still computed from the entire image contents (figure* 6(a, d)) and are still segmented in the same way to provide cuts with an equal number of regions. figure* 7(a) presents a typical example of a urban area composed of structured complex urban blocks. In this example, we observe that the segmentation result provided by the "standard" BPT (figure* 7(b)) does not enable to reconstruct

(a) Image crop (4). (b) BPT. (c) MBPT-1. (d) MBPT-2.

Figure 10: Segmentation results from the BPTs and the MBPTs, centred on crop (4) of the image presented in figure* 6(a). BPT: $\{W_{colour}\}$; MBPT-1: $\{W_{colour}, W_{ndvi}\}$; MBPT-2: $\{W_{colour}, W_{elong}, W_{ndvi}\}$.

these urban blocks since they often appear composed of several small sub-regions (see in particular the road segments). By adding the valuation function W_{ndvi} in the creation process of our multi-criteria BPT (figure* 7(c)), we observe that the extracted cut permits to regroup connected regions having similar NDVI values (objects containing vegetation pixels or not). In this result, the garden areas between houses (see centre bottom of the crop (1)) are gathered in larger segments and the road sections are better connected than before. This last phenomenon is also easily observable in figure* 7(d) where the cut was performed on a multi-criteria BPT involving four valuation functions $\{W_{colour}, W_{elong}, W_{ndvi}, W_{ndwi}\}$. In this example, the elongation criterion permitted to produce compact and large road sections.

In figure* 8, it can be noticed that the river, at the right side of the crop (2), is divided by three segments in the result obtained from the "standard" BPT figure* 8(b). By considering the W_{ndwi} valuation function during the MBPT creation process (see segmentation result in figure* 8(c)), the river corresponds to only one segment. Indeed, this valuation function enables to quantify the similarity between neighbouring regions containing water or not. We also observe that the segments of road are also less disconnected since they do not contain water but they present similar (low) NDWI values. The best result for the segmentation of both simple and complex objects is shown in figure* 8(d) where four valuation functions are involved $\{W_{colour}, W_{elong}, W_{ndvi}, W_{ndwi}\}$. Here, the road sections are well delineated, the river belongs to one segment and the vegetation area (at the right side of the crop (2)) looks more homogeneous. Our assumption is that complex urban objects appearing in the image content as either 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 MBPT creation.

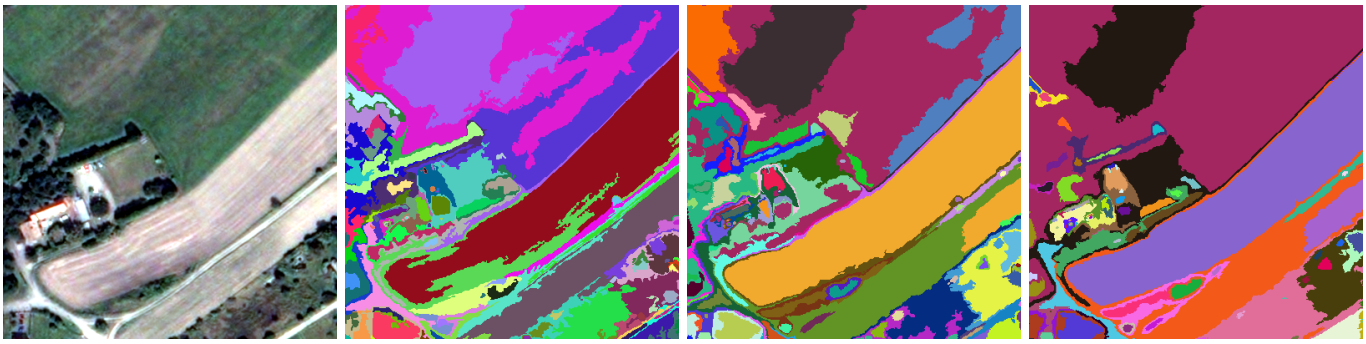
In comparison to what we observe in figure* 9(b), which is a result from the "standard" BPT, the use of the W_{ndvi} and the W_{ndwi} valuation functions together in figure* 9(c) enhances the segmentation of both the vegetation areas and the bare soils in an urban area. Adding the W_{elong} metric in figure* 9(d) also helped the creation of the curved road and the small elongated group of trees at the right side of the crop (3). However, some geographical objects having certainly similar NDVI and / or NDWI values are mixed up together although they have different semantics. Indeed, the building roofs at the right side of the crop (3) are merged with the parking lot.

The same phenomenon is observed in figure* 10(c, d) where the triangular bare soil, the road sections and the vegetation areas are visually better segmented with the MBPTs than with the "standard" BPT (see figure* 10(b)).

The image crops extracted from the low-density urban area image (figure* 6(d)) represent illustrative examples of vegetation areas, rivers and scattered urban objects. figure* 11(c) and figure* 12(c) highlight the ability of the W_{ndvi} valuation function to regroup objects containing live green vegetations. In these examples the wide vegetation areas are gathered in larger segments while they were divided in several sub-regions in the results obtained from the "standard" BPTs (see figure* 11(b) and figure* 12(b)). The consensual results obtained from the use of more valuation functions in the MBPTs (see figure* 11(d) and figure* 12(d)) enable to deal with the colour heterogeneity between the different crop fields.

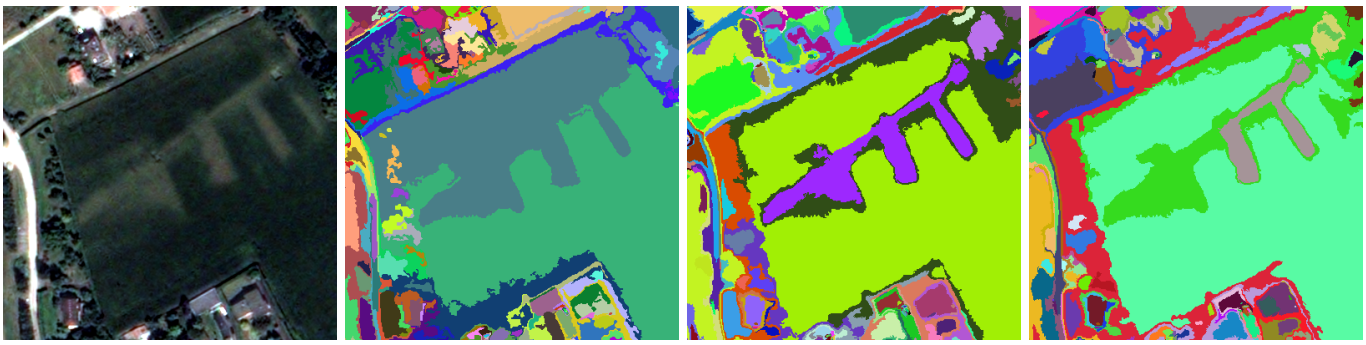
figure* 13(a) represents an area composed of urban objects mixed up with some complex vegetation zones. The road section are visible in the result from the "standard" BPT (see figure* 13(b)) but the vegetation areas at the right side of the crop (7) are divided in numerous small regions. From the results of the multi-feature BPT involving both the W_{colour} and the W_{ndvi} valuation functions (figure* 13(c)), we observe that the vegetation areas and the road sections form now better connected regions. The result obtained from adding the W_{elong} metric to the MBPT (figure* 13(d)) presents a balanced consensual segmentation where road sections can be easily recognized and the vegetation areas are more homogeneous.

Finally, the result obtained from the "standard" BPT,



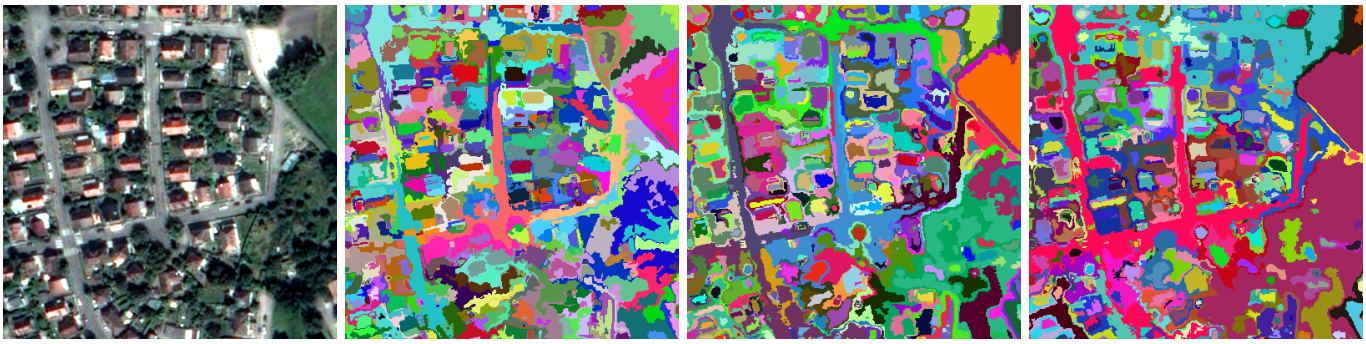
(a) Image crop (5). (b) BPT. (c) MBPT-1. (d) MBPT-2.

Figure 11: Segmentation results from the BPTs and the MBPTs, centred on crop (5) of the image presented in figure* 6(d). BPT: $\{W_{colour}\}$; MBPT-1: $\{W_{colour}, W_{ndvi}\}$; MBPT-2: $\{W_{colour}, W_{elong}, W_{ndvi}, W_{ndwi}\}$.



(a) Image crop (6). (b) BPT. (c) MBPT-1. (d) MBPT-2.

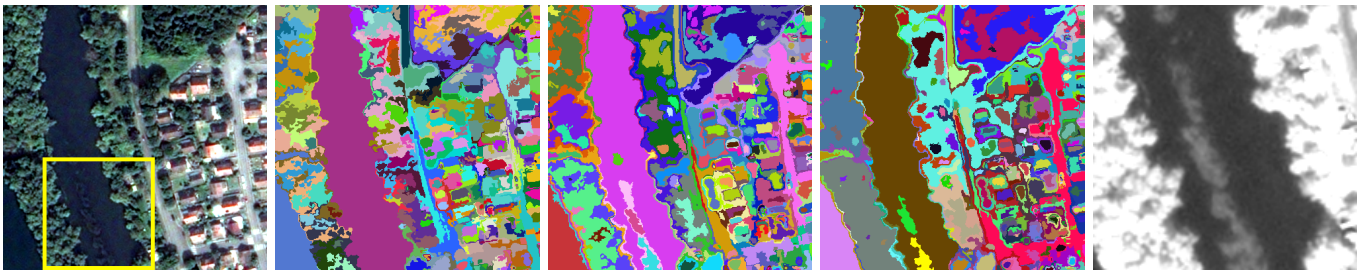
Figure 12: Segmentation results from the BPTs and the MBPTs, centred on crop (6) of the image presented in figure* 6(d). BPT: $\{W_{colour}\}$; MBPT-1: $\{W_{colour}, W_{ndvi}\}$; MBPT-2: $\{W_{colour}, W_{ndvi}, W_{ndwi}\}$.



(a) Image crop (7). (b) BPT. (c) MBPT-1. (d) MBPT-2.

Figure 13: Segmentation results from the BPTs and the MBPTs, centred on crop (7) of the image presented in figure* 6(d). BPT: $\{W_{colour}\}$; MBPT-1: $\{W_{colour}, W_{ndvi}\}$; MBPT-2: $\{W_{colour}, W_{elong}, W_{ndvi}, W_{ndwi}\}$.

in figure* 14(b), shows that the vegetation areas at the left part of the crop (8) are divided in several small sub-regions, the river seems to be perfect and the urban zone is over-segmented. In comparison, the result obtained from a multi-feature BPT using the W_{ndwi} valuation function offers an unexpected observation. The river, that seems to be totally homogeneous in figure* 14(a), is segmented differently in figure* 14(c, d). In fact, the result obtained for the multi-feature BPTs helped to highlight some vegetation structures that cannot be easily interpreted by human observation. Indeed, the river is composed of small sand-banks covered by vegetation tissues that could be clearly observed when we focus on the near infrared band (see the zoom in the figure* 14(e)). By combining four valuation



(a) Img. crop (8). (b) BPT. (c) MBPT-1. (d) MBPT-2. (e) NIR band.

Figure 14: Segmentation results from the BPTs and the MBPTs, centred on crop (8) of the image presented in figure* 6(d). BPT: $\{W_{colour}\}$; MBPT-1: $\{W_{colour}, W_{ndwi}\}$; MBPT-2: $\{W_{colour}, W_{elong}, W_{ndvi}, W_{ndwi}\}$.

functions $\{W_{colour}, W_{elong}, W_{ndvi}, W_{ndwi}\}$ in figure* 14(d), a consensual segmentation result shows us a pertinent balance between the vegetation areas that are quite homogeneous, the urban area where the road sections are well structured, and the complex content of the river that we observed earlier.

7.2. Illustrative example 2: Multi-image segmentation

We now illustrate the interest of multi-criteria BPT for multi-image segmentation in the context of satellite imaging. It has to be noticed that involving BPTs in a multi-image processing context is innovative since to the best of our knowledge BPTs have never been used for this purpose.

7.2.1. Data

The dataset used here is a time series of images (1000 × 1000 pixels) sensed over an area located near Toulouse (France). This area is a typical agricultural zone composed of different types of crop fields and vegetations. Images were acquired by the FORMOSAT-2 satellite over the 2007 cultural year. They were ortho-rectified and have a spatial resolution of 8 m, with four spectral bands (R, G, B, NIR). From this dataset, we selected four images (Figure 15) all acquired late August and September to reduce as much as we can the temporal evolution effects. Note that the second image is partially affected by the presence of clouds. The main purpose of this experiment is actually to assess the ability of our multi-feature BPT framework to capture time-independent and redundant information from the contents of multiple images representing the same scene.

7.2.2. Method and results

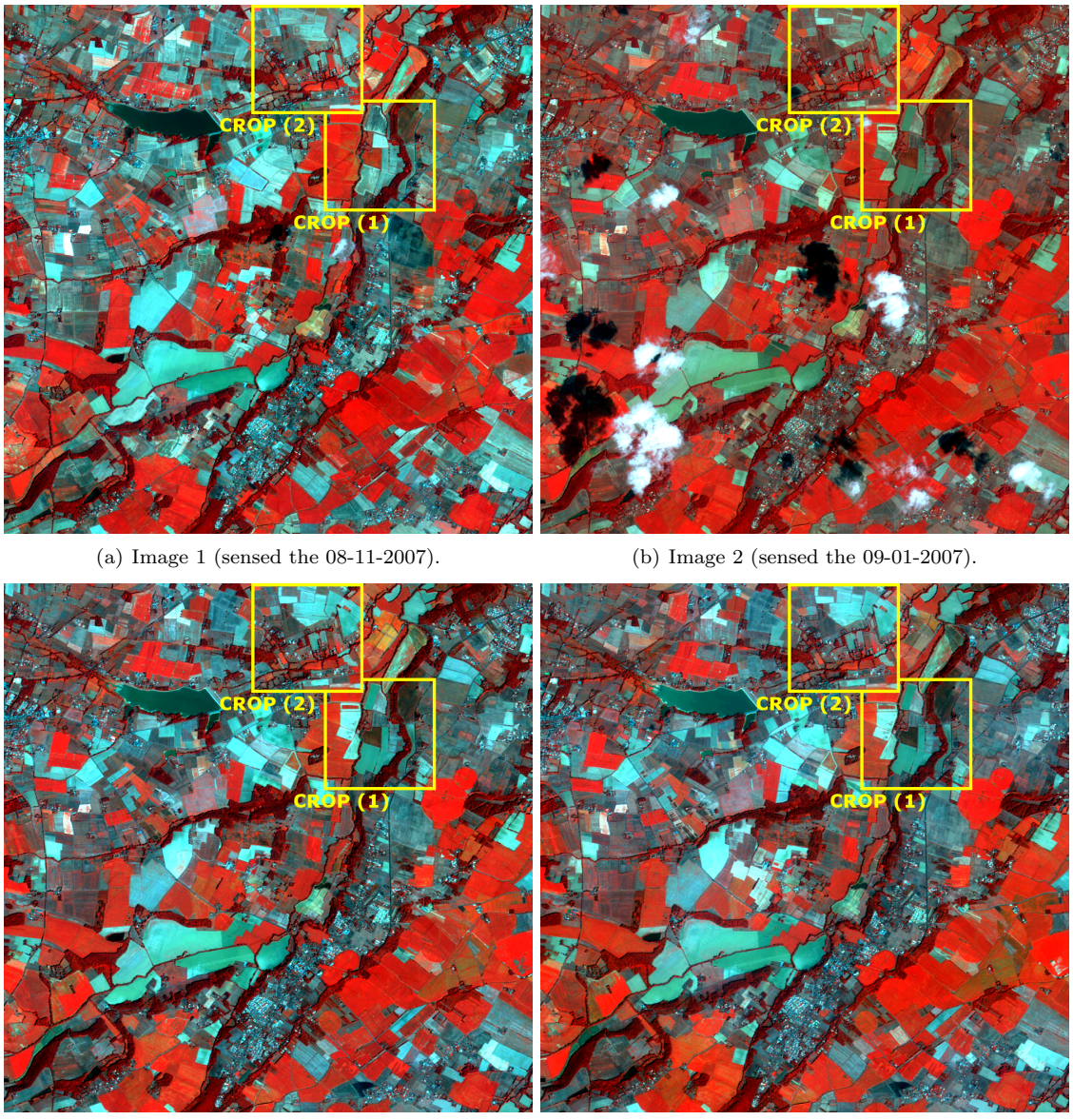
To reduce the spatial complexity of this approach, the BPTs are built from an initial partition $\mathcal{L}$ composed of 200000 regions obtained from a cut performed on a "standard" BPT. This initial partition was produced from the Image 1 (figure* 15(a)) that is not affected by the presence of cloud.

In the case of multi-image 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 15% of the lists $\mathcal{W}_\star$.

As a baseline, a "standard" BPT is construct for each satellite image presented in figure* 15 by considering the intensity value criterion W_{colour} (denoted hereinafter as W_{colour}^{img1}, W_{colour}^{img2}, W_{colour}^{img3}, W_{colour}^{img4} depending on the image where this criterion is computed). These binary trees are then segmented by using a user-defined horizontal cut to produce a partition whose the region scales are adapted to segment the various geographical objects covering the sensed areas (e.g., agricultural crop fields, wide forest areas), see figure* 16(f, g, h, i) and figure* 17(f, g, h, i). For visualisation purpose, the segmentation results are depicted here in random false colours on illustrative image crops.

To evaluate the impact of the different valuation functions on the segmentation results, the multi-image BPTs are built by considering simultaneously informations extracted from the contents of the four satellite images (e.g., $\{W_{colour}^{img1}, \ldots, W_{colour}^{img3}\}$, $\{W_{colour}^{img1}, W_{colour}^{img2}, W_{ndvi}^{img1}, W_{ndvi}^{img2}\}$). The produced multi-image trees are then segmented in the same way as for the "standard" BPTs, leading to the same number of regions.

For a better visualization, we focus in the following on two image crops representing different parts of the geographical scene. In the crop (1) represented by figure* 16(a, b, c, d), we observe that, although some variations are due to the time evolutions, the main structures of the crop fields do not evolve. After a visual interpretation of these images, it can be noticed that the majority of observed geographical areas preserve the same crop field partition. However, each segmentation obtained from the "standard" BPT of each image presents totally different partitions (see figure* 16(f, g, h, i)). Indeed, some particular crop fields (centre of crop (1)) are always divided in various number of regions (sometimes few and sometimes more). The same phenomenon can be observed with the elongated vegetation area that is sometimes merged with the neighbouring crop fields. In comparison, the results obtained from our multi-image BPT (see figure* 16(e)) are more stable since they arise from a consensual discussion between the results of the "standard" BPT results.



(a) Image 1 (sensed the 08-11-2007).

(b) Image 2 (sensed the 09-01-2007).

(c) Image 3 (sensed the 09-08-2007).

(d) Image 4 (sensed the 09-15-2007).

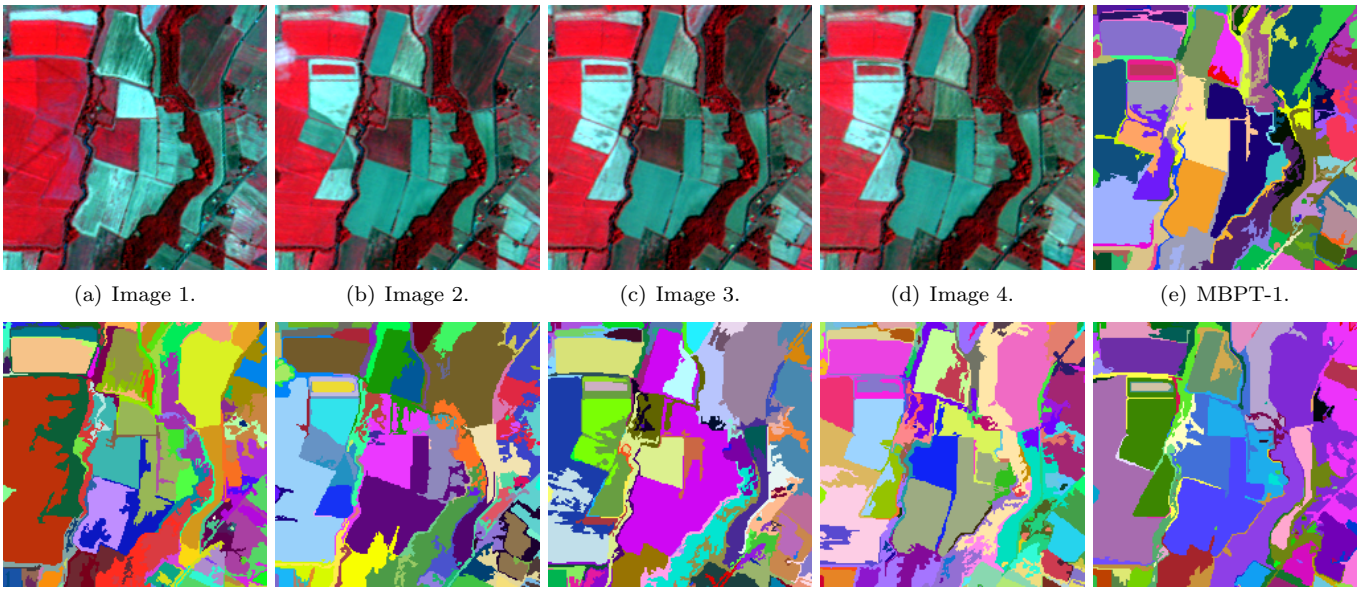
Figure 15: Illustration of four satellite images (1000 × 1000 pixels) at a spatial resolution of 8 m sensed by the FORMOSAT-2 satellite covering the same geographical area.

The same behaviour can be observed in the second crop (figure* 17) where the results obtained from the "standard" BPT of each image (see figure* 17(f, g, h, i)) are almost totally different but most important information are gathered in the result of our multi-image BPT (see figure* 17(e)).

The best results are shown in figure* 16(j) and figure* 17(j) where we used three of the four images with two metrics $\{W^{img1}_{colour}, \ldots, W^{img3}_{colour}, W^{img1}_{ndvi}, \ldots, W^{img3}_{ndvi}\}$. The regions corresponding to crop fields are more homogeneous and the number of sub-regions of each area is well adapted for a future object-based recognition or classification step.

8. Conclusion

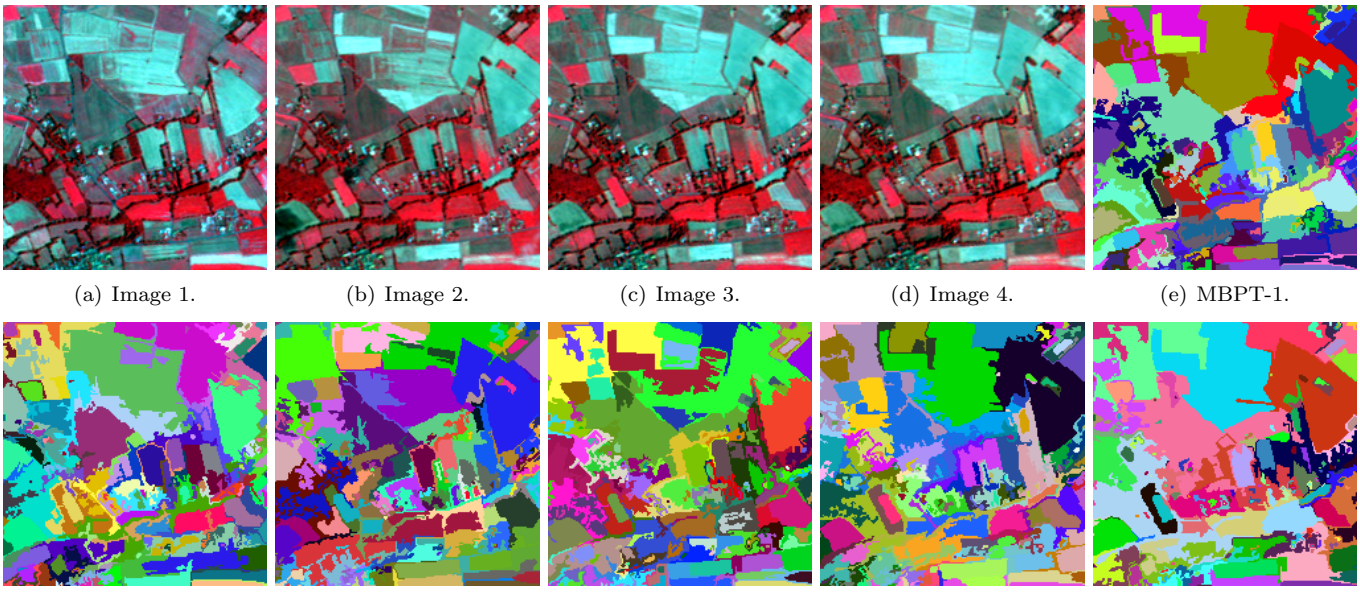
In this article, we proposed a generalization of the BPT construction framework, classically built in a mono-feature way, thus allowing to consider various multi-feature paradigms. Such a multi-feature framework enables to reduce the task of the user, by offering more flexibility for the BPT creation. 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



(a) Image 1. (b) Image 2. (c) Image 3. (d) Image 4. (e) MBPT-1.

(f) BPT-1. (g) BPT-2. (h) BPT-3. (i) BPT-4. (j) MBPT-2.

Figure 16: Segmentation results from the mono-image BPTs and the multi-images/metrics MBPTs. (a) Crop 1 from Image 1 (Figure 15(a)). (b) Crop 1 from Image 2 (Figure 15(b)). (c) Crop 1 from Image 3 (Figure 15(c)). (d) Crop 1 from Image 4 (Figure 15(d)). (f) Seg. result from a BPT of Image 1. (g) Seg. result from a BPT of Image 2. (h) Seg. result from a BPT of Image 3. (i) Seg. result from a BPT of Image 4. (e) Seg. result from a MBPT with 4-images/1-metric $\{W_{colour}^{img1}, W_{colour}^{img2}, W_{colour}^{img3}, W_{colour}^{img4}\}$. (j) Seg. result from a MBPT with 3-images/2-metrics $\{W_{colour}^{img1}, \ldots, W_{colour}^{img3}, W_{ndvi}^{img1}, \ldots, W_{ndvi}^{img3}\}$.



(a) Image 1. (b) Image 2. (c) Image 3. (d) Image 4. (e) MBPT-1.

(f) BPT-1. (g) BPT-2. (h) BPT-3. (i) BPT-4. (j) MBPT-2.

Figure 17: Segmentation results from the mono-image BPTs and the multi-images/metrics MBPTs. (a) Crop 2 from Image 1 (Figure 15(a)). (b) Crop 2 from Image 2 (Figure 15(b)). (c) Crop 2 from Image 3 (Figure 15(c)). (d) Crop 2 from Image 4 (Figure 15(d)). (f) Seg. result from a BPT of Image 1. (g) Seg. result from a BPT of Image 2. (h) Seg. result from a BPT of Image 3. (i) Seg. result from a BPT of Image 4. (e) Seg. result from a MBPT with 4-images/1-metric $\{W_{colour}^{img1}, W_{colour}^{img2}, W_{colour}^{img3}, W_{colour}^{img4}\}$. (j) Seg. result from a MBPT with 3-images/2-metrics $\{W_{colour}^{img1}, \ldots, W_{colour}^{img3}, W_{ndvi}^{img1}, \ldots, W_{ndvi}^{img3}\}$.

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 framework, the short-term perspective of this work will be to implement distributed heuristics based on graph-based distribution algorithmic. 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, M. H. F. Wilkinson, Connected operators: A review of region-based morphological image processing techniques, IEEE Signal Processing Magazine 26 (2009) 136–157.

[2] L. Najman, J. Cousty, A graph-based mathematical morphology reader, Pattern Recogntion Letters 47 (2014) 3–17.

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

[4] P. Salembier, L. Garrido, Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval, IEEE Transactions on Image Processing 9 (2000) 561–576.

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

[6] A. Rosenfeld, Connectivity in digital pictures, Journal of the ACM 17 (1970) 146–160.

[7] R. Adams, L. Bischof, Seeded region growing, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (1994) 641–647.

[8] L. Vincent, P. Soille, Watersheds in digital spaces: An efficient algorithm based on immersion simulations, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (1991) 583–598.

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

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

[11] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 1222–1239.

[12] L. Grady, Random walks for image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006) 1768–1783.

[13] C. Couprie, L. Grady, L. Najman, H. Talbot, Power watershed: A unifying graph-based optimization framework, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (2011) 1384–1398.

[14] P. Salembier, J. Serra, Flat zones filtering, connected operators, and filters by reconstruction, IEEE Transactions on Image Processing 4 (1995) 1153–1160.

[15] M.-M. Yau, S. N. Srihari, A hierarchical data structure for multidimensional digital images, Communications of the ACM 26 (1983) 504–515.

[16] A. Montanvert, P. Meer, A. Rosenfeld, Hierarchical image analysis using irregular tessellations, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (1991) 307–316.

[17] P. Salembier, A. Oliveras, L. Garrido, Anti-extensive connected operators for image and sequence processing, IEEE Transactions on Image Processing 7 (1998) 555–570.

[18] P. Monasse, F. Guichard, Scale-space from a level lines tree, Journal of Visual Communication and Image Representation 11 (2000) 224–236.

[19] P. Monasse, F. Guichard, Fast computation of a contrast-invariant image representation, IEEE Transactions on Image Processing 9 (2000) 860–872.

[20] L. Najman, M. Schmitt, Geodesic saliency of watershed contours and hierarchical segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (1996) 1163–1173.

[21] B. Perret, S. Lefèvre, C. Collet, E. Slezak, Hyperconnections and hierarchical representations for grayscale and multiband image processing, IEEE Transactions on Image Processing 21 (2012) 14–27.

[22] R. Jones, Connected filtering and segmentation using component trees, Computer Vision and Image Understanding 75 (1999) 215–228.

[23] L. Guigues, J.-P. Cocquerez, H. Le Men, Scale-sets image analysis, International Journal of Computer Vision 68 (2006) 289–317.

[24] N. Passat, B. Naegel, F. Rousseau, M. Koob, J.-L. Dietemann, Interactive segmentation based on component-trees, Pattern Recognition 44 (2011) 2539–2554.

[25] N. Passat, B. Naegel, Component-trees and multivalued images: Structural properties, Journal of Mathematical Imaging and Vision 49 (2014) 37–50.

[26] C. Kurtz, B. Naegel, N. Passat, Connected filtering based on multivalued component-trees, IEEE Transactions on Image Processing 23 (2014) 5152–5164.

[27] E. Carlinet, T. Géraud, MToS: A tree of shapes for multivariate images, IEEE Transactions on Image Processing 24 (2015) 5330–5342.

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

[29] B. Perret, J. Cousty, O. Tankyevych, H. Talbot, N. Passat, Directed connected operators: Asymmetric hierarchies for image filtering and segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (2015) 1162–1176.

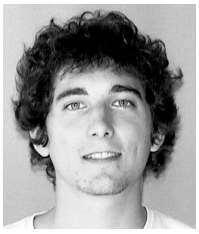
[30] Y. Xu, T. Géraud, L. Najman, Connected filtering on tree-based shape-spaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (2016) 1126–1140.

[31] É. Grossiord, B. Naegel, H. Talbot, N. Passat, L. Najman, Shape-based analysis on component-graphs for multivalue image processing, in: ISMM, International Symposium on Mathematical Morphology, Proceedings, volume 9082 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 446–457.

[32] J. L. Bentley, Multidimensional binary search trees used for associative searching, Communications of the ACM 18 (1975)

509–517.
[33] P. Soille, Constrained connectivity for hierarchical image partitioning and simplification, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 1132–1145.
[34] S. Valero, P. Salembier, 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, F. Marqués, From partition trees to semantic trees, in: MRCS, Multimedia Content Representation, Classification and Security, Proceedings, volume 4105 of *Lecture Notes in Computer Science*, Springer, 2006, pp. 306–313.
[36] V. Vilaplana, F. Marques, P. Salembier, Binary partition trees for object detection, IEEE Transactions on Image Processing 17 (2008) 2201–2216.
[37] S. Valero, P. Salembier, J. Chanussot, Object recognition in hyperspectral images using binary partition tree representation, Pattern Recognition Letters 56 (2015) 45–51.
[38] J. A. Benediktsson, L. Bruzzone, J. Chanussot, M. Dalla Mura, P. Salembier, S. Valero, Hierarchical analysis of remote sensing data: Morphological attribute profiles and binary partition trees, in: ISMM, International Symposium on Mathematical Morphology, Proceedings, Lecture Notes in Computer Science, Springer, 2011, pp. 306–319.
[39] C. Kurtz, N. Passat, P. Gançarski, A. Puissant, Extraction of complex patterns from multiresolution remote sensing images: A hierarchical top-down methodology, Pattern Recognition 45 (2012) 685–706.
[40] F. Calderero, F. Eugenio, J. Marcello, F. Marques, Multispectral cooperative partition sequence fusion for joint classification and hierarchical segmentation, IEEE Geoscience and Remote Sensing Letters 9 (2012) 1012–1016.
[41] C. Kurtz, A. Stumpf, J.-P. Malet, P. Gançarski, A. Puissant, N. Passat, Hierarchical extraction of landslides from multiresolution remotely sensed optical images, ISPRS Journal of Photogrammetry and Remote Sensing 87 (2014) 122–136.
[42] S. Valero, P. Salembier, J. Chanussot, Hyperspectral image representation and processing with binary partition trees, IEEE Transactions on Image Processing 22 (2013) 1430–1443.
[43] M. A. Veganzones, G. Tochon, M. Dalla Mura, A. J. Plaza, J. Chanussot, Hyperspectral image segmentation using a new spectral unmixing-based binary partition tree representation, IEEE Transactions on Image Processing 23 (2014) 3574–3589.
[44] A. Alonso-González, C. López-Martínez, P. Salembier, Filtering and segmentation of polarimetric SAR data based on binary partition trees, IEEE Transactions on Geoscience and Remote Sensing 50 (2012) 593–605.
[45] P. Salembier, Study of binary partition tree pruning techniques for polarimetric SAR images, in: ISMM, International Symposium on Mathematical Morphology, Proceedings, volume 9082 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 51–62.
[46] A. Alonso-González, S. Valero, J. Chanussot, C. López-Martínez, P. Salembier, Processing multidimensional SAR and hyperspectral images with binary partition tree, Proceedings of the IEEE 101 (2013) 723–747.
[47] A. Alonso-González, C. López-Martínez, P. Salembier, PolSAR time series processing with binary partition trees, IEEE Transactions on Geoscience and Remote Sensing 52 (2014) 3553–3567.
[48] L. Garrido, P. Salembier, D. Garcia, Extensive operators in partition lattices for image sequence analysis, Signal Processing 66 (1998) 157–180.
[49] C. Kurtz, N. Passat, A. Puissant, P. Gançarski, Hierarchical segmentation of multiresolution remote sensing images, in: ISMM, International Symposium on Mathematical Morphology, Proceedings, volume 6671 of *Lecture Notes in Computer Science*, Springer, 2011, pp. 343–354.
[50] J. Cousty, L. Najman, B. Perret, Constructive links between some morphological hierarchies on edge-weighted graphs, in: ISMM, International Symposium on Mathematical Morphology, Proceedings, volume 7883 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 86–97.
[51] A. Al-Dujaili, F. Merciol, S. Lefèvre, GraphBPT: An efficient hierarchical data structure for image representation and probabilistic inference, in: ISMM, International Symposium on Mathematical Morphology, Proceedings, volume 9082 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 301–312.
[52] V. Machairas, M. Faessel, D. Cárdenas-Peña, T. Chabardes, T. Walter, E. Decencière, Waterpixels, IEEE Transactions on Image Processing 24 (2015) 3707–3716.

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.