Supervised quality evaluation of binary partition trees for object segmentation

Jimmy Tianatahina Francky Randrianasoa, Pierre Cettour-Janet, Camille Kurtz, Eric Desjardin, Pierre Gançarski, Nathalie Bednarek, François Rousseau, Nicolas Passat

Supervised quality evaluation of binary partition trees for object segmentation

Jimmy Francky Randrianasoa[a], Pierre Cettour-Janet[b], Camille Kurtz[c], Éric Desjardin[b], Pierre Gançarski[d], Nathalie Bednarek[b,e], François Rousseau[f], Nicolas Passat[b]

[a] *Télécom Saint-Étienne, CNRS, Laboratoire Hubert Curien, Saint-Étienne, France*
[b] *Université de Reims Champagne Ardenne, CReSTIC EA 3804, 51097 Reims, France*
[c] *Université de Paris, LIPADE, Paris, France*
[d] *Université de Strasbourg, CNRS, ICube, Strasbourg, France*
[e] *Service de médecine néonatale et réanimation pédiatrique, CHU de Reims, France*
[f] *IMT Atlantique, LaTIM U1101 INSERM, UBL, Brest, France*

Abstract

The binary partition tree (BPT) allows for the hierarchical representation of images in a multiscale way, by providing a tree of nodes corresponding to image regions. In particular, cuts of a BPT can be interpreted as segmentations of the associated image. Building the BPT of an image then constitutes a relevant preliminary step for optimization-based segmentation methods. A wide literature has been devoted to the construction of BPTs, and their involvement in such segmentation tasks. Comparatively, there exist few works dedicated to evaluate the quality of BPTs, i.e. their ability to allow further segmentation methods to compute good results. We propose such a framework for evaluating the quality of a BPT with respect to the object segmentation problem, i.e. the segmentation of one or several objects from an image. This framework is supervised, since the notion of segmentation quality is not only depending on the application but also on the user's objectives, expressed via the chosen ground-truth and quality metric. We develop two sides within this framework. First, we propose an intrinsic quality analysis, that relies on the structural coherence of the BPT with respect to ground-truth. More precisely, we evaluate to what extent the BPT structure is well-matching such examples, in a set / combinatorial fashion. Second, we propose an extrinsic analysis, by allowing the user to assess the quality of a BPT based on chosen metrics that correspond to the desired properties of the subsequent segmentation. In particular, we evaluate to what extent a BPT can provide good results with respect to such metrics whereas handling the trade-off with the cardinality of the cuts.

Keywords: Binary partition tree, object segmentation, hierarchical image

model, supervised quality evaluation, mathematical morphology

1. Introduction

Carrying out segmentation on an image, which basically relies on the computation a partition of its support, is a challenging task. This is mainly due to the super-exponential number of possible partitions that exist for a given set of pixels. In order to tackle this combinatorial issue, a classical solution consists of defining—before the segmentation process—a lower-size subspace of candidate partitions, within the whole research space of all the possible partitions.

The most popular way of defining such reduced partition spaces is to build hierarchies of partitions. The purpose is then to pre-compute some image regions whose union corresponds to non-overlapping covers of the image support. This often leads to graph-based data structures organized as trees (i.e. rooted, connected, acyclic graphs) that model partitions hierarchically organized with respect to the refinement relation.

Since the pioneering works on tree structures for image processing and analysis (i.e. regular [1] and non-regular pyramids [2]), various kinds of hierarchies of partitions have been proposed. Many of them were developed in the framework of mathematical morphology [3]. These hierarchies are mostly *intrinsic* image models; this means that the information required for building a tree from an image is contained within the image. This is, for instance, the case of component-trees [4] and multivalued component-trees [5] that rely on level-sets; trees of shapes [6] and multivariate trees of shapes [7] based on isocontours; hierarchical watersheds [8] (that rely also on a chosen gradient policy which constitutes, in some way, an extrinsic meta-parameter); hyperconnection trees [9], etc.

Some *mixed* hierarchies of partitions were also proposed. The most popular are the binary partition trees (BPTs) [10] and their variants (see e.g. [11, 12]). BPTs can be considered as mixed hierarchies since they rely not only on the information directly embedded in the image, but also on extrinsic information, namely prior knowledge related to sought structures of interest, modelled via a metric. Then, both image and metric are involved in the construction of the BPT data structure.

When dealing with an intrinsic image model, the quality of the segmentation result only depends on the efficiency of the segmentation algorithm, i.e. the strategy for defining a relevant cut within the tree. By contrast, when deal-

ing with a BPT (and, more generally, a mixed image model), the quality of the segmentation result depends not only on the efficiency of the segmentation algorithm, but also on the quality of the hierarchy of partitions built beforehand. In other words, when considering BPT-based image processing, *a good segmentation algorithm has to be applied on a good image tree data structure.*

A wide literature has been devoted to BPT-based segmentation, mostly in the challenging context of remote sensing [13, 14, 15, 16, 17, 18, 19, 20, 21, 22]. Various metrics were investigated (spectral, spatial, geometric, etc.) for building BPTs. The design of these metrics (choice and combination) strongly influences the resulting BPTs, i.e. the reduced search space for further segmentation, and thus the quality of the subsequent segmentation result.

Comparatively to the rich panel of contributions geared towards building BPTs and using them for segmentation purposes, the literature dedicated to evaluating the quality of BPTs is much sparser. Indeed, (binary partition) trees are complex objects that model sets of partitions. Then, they are structurally more difficult and spatially more costly to analyze compared to single segmentations which correspond to unique partitions [23, 24, 25, 26, 27, 28].

The contributions dedicated to BPT evaluation in the context of segmentation can be subdivided in two families, that correspond to two paradigms of segmentation. Indeed, the segmentation of an image is defined from the partition of its support. On the one hand, this partition can be seen as the very segmentation. In such case, the purpose is to maximize (resp. minimize) intra-region (resp. inter-region) homogeneity with respect to given (spectral, semantic...) properties. This is for instance the case in scene labelling in computer vision, or in image classification in remote sensing. On the other hand, the partition can be used only for identifying some objects of interest. In such context, an object is defined by one region of the partition or the union of different regions whereas all other regions are considered non-relevant. In other words, we are only interested in parts of the image formed from unions of regions (namely the foreground, or the object) versus the union of the remaining regions (namely the background). In that case, we will speak about *object* segmentation, and the final purpose is to interpret a given partition as a coarser binary partition. This is for instance the case in medical imaging, when segmenting anatomical structures (e.g. tumour lesions in an organ) or in remote sensing when looking for specific objects (e.g. red-roof buildings).

Regarding the evaluation frameworks dedicated to the first kind of segmentations, that correspond to whole partitions of the image, the considered ground-

truth is itself a partition, and the quality of the tree is related to the closeness of its best induced partition with respect to this ground-truth. In this context, the main quality scores considered are the precision, the recall and the F-measure (their harmonic mean). This is for instance the case in [29], where these metrics are assessed by considering a subset of (horizontal) cuts, which number is linear with respect to the size of the tree. In order to go beyond the case of horizontal cuts (which are not necessarily those likely to provide the more representative segmentation results), two approaches were proposed, that rely on assumptions on the considered quality metrics, namely the ability of the optimal partition of the image to be defined as the union of the optimal partitions of the two half-images [30]; and the ability of the optimal cut to be the solution of a linear fractional combinatorial optimization problem, which is the case for the F-measure for boundary detection that relies on the precision/recall trade-off [31]. A recent framework gathering these precision/recall-based strategies can be found in [32].

Regarding the evaluation frameworks dedicated to the second kind of segmentation, that corresponds to a binary partition of the image, to our best knowledge, the closest work is the one proposed in [33] (see also [32]), which is mainly geared towards evaluating the quality of trees with respect to foreground / background markers. By contrast to the first category of evaluation frameworks, the ground-truth considered in this second family are more simple, since they are binary objects. In this context, the problem to be solved is no longer about comparing how much two partitions are similar, but determining how much a partition allows one to build a binary object. This requires, on the one hand, to handle user-defined quality metrics that may be different from precision and recall. On the other hand, this also requires to assess the partitions in a more intrinsic way since a "good" partition is not only a partition which union of subsets leads to a fairly correct segmented object, but also a partition with a sufficiently low number of elements and composed of regions individually well-matching the object of interest. In the sequel, we propose a framework for BPT quality evaluation in this context of object segmentation.

In the proposed framework, we allow the user to assess the quality of a BPT based on chosen quality metrics that correspond to the desired properties of a subsequent segmentation, but also by considering combinatorial properties of the tree. More precisely, we evaluate to what extent some cuts of a BPT can optimize such quality metrics, whereas handling the trade-off between the metric and the size of the cut (with the assumption that the lower the size, the

better the segmentation, with the best case where the segmentation result is indeed given by exactly one node of the BPT).

This article is an extended and improved version of the conference articles [34, 35, 36] which dealt either with intrinsic or extrinsic analysis frameworks dedicated to object segmentation from BPTs. The intrinsic framework introduced in [35] is recalled with additional details, whereas a new extrinsic framework is now proposed, building upon the preliminary approaches introduced in [34, 36]. An homogeneous formalism is now proposed in order to make the overall framework more easily comprehensible. A new experimental section compares the results of this framework with other frameworks mainly dedicated to partition-based segmentation but applicable, under appropriate adaptations, to the case of object segmentation.

The remainder of this article is organized as follows. In Section 2, we introduce notations and we recall background notions related to BPTs, required to make this article self-contained. In Section 3, we describe the problem that we aim to tackle, namely the evaluation of the quality of a BPT as an efficient data structure for carrying out segmentation via tree-cut computation. Sections 4–5 describe the proposed evaluation framework and constitute the core of this article. Section 6 proposes an experimental study dedicated to illustrate the behaviour of the proposed framework in the context of natural image segmentation, and compare this framework to the closest ones of the literature. Section 7 concludes this article by emphasizing perspective works.

2. Background Notions

A list of the principal notations is provided in Appendix.

2.1. Image

An image I is defined on a nonempty, finite set of points Ω. In general, such points are pixels (resp. voxels) in 2D (resp. 3D) images. The structural properties between these points have to be explicitly considered. Practically, the relevant information is the existence of a neighbourhood relation between two points. We can model this information as an adjacency (i.e. irreflexive, symmetric) relation on Ω, noted $\frown$. Then, the physical support of an image I will be considered as the graph $(\Omega, \frown)$.

The image I takes its values in a space $\mathbb{V}$. There is no actual requirement with regard to this value space. In general, $\mathbb{V}$ is a scalar space (e.g. a set of

real or integer values) in the case of monovalued images, or a Cartesian set of such spaces in the case of multi-/hyperspectral images. Hypotheses on $\mathbb{V}$ have no impact on the very structure of a BPT. They can, however, be involved in the parametrization of its construction process.

Practically, an image can then be formalized as a function $I : \Omega \to \mathbb{V}$. For BPT construction and handling issues, we will mainly consider its spatial support Ω and its graph-based structure $(\Omega, \frown)$.

2.2. Binary Partition Tree

A BPT associated to $(\Omega, \frown)$ is a tree $\mathfrak{T} = (\mathcal{N}, \searrow)$, i.e. a directed, connected, acyclic graph. In particular, the set $\mathcal{N}$ which is a part of the power-set of Ω is the union of three subsets:

- $\mathcal{R} = \{\Omega\}$ (we will note $N_\Omega = \Omega$);
- $\mathcal{B}$;
- $\mathcal{L} = \{\{x\}\}_{x \in \Omega}$ (we will note $N_x = \{x\}$);

such that for any $N \in \mathcal{R} \cup \mathcal{B}$ (resp. $\mathcal{L}$), there exist exactly two (resp. no) $N' \in \mathcal{N}$ satisfying $N \searrow N'$ and N is the union of these two elements; and for any $N \in \mathcal{B} \cup \mathcal{L}$ (resp. $\mathcal{R}$), there exists exactly one (resp. no) $N' \in \mathcal{N}$ satisfying $N' \searrow N$. In such conditions, we have in particular $|\mathcal{R}| = 1$, $|\mathcal{L}| = |\Omega|$, $|\mathcal{B}| = |\Omega| - 2$, and $|\mathcal{N}| = 2|\Omega| - 1$.

Less formally, a BPT $\mathfrak{T}$ of $(\Omega, \frown)$ provides a family $\mathcal{N}$ of subsets of Ω. These subsets are hierarchically organized from the whole set Ω to the singleton sets $\{x\}$, $x \in \Omega$, with respect to the inclusion relation. This hierarchical organization is characterized by the fact that an element N of $\mathcal{N}$, which is not a singleton set, is associated via $\searrow$ to exactly two elements N_1, N_2 of $\mathcal{N}$ that form a binary partition of N.

We use the standard terminology of trees: the elements of $\mathcal{N}$ are the nodes of $\mathfrak{T}$, the node Ω is the root of $\mathfrak{T}$, the singleton nodes of $\mathcal{L}$ are the leaves of $\mathfrak{T}$.

By construction, for any two nodes of $\mathcal{N}$, the intersection between them implies that one is included in the other.

2.3. Construction of a Binary Partition Tree

A BPT $\mathfrak{T}$ can be built from a graph $(\Omega, \frown)$ either in a top-down or a in bottom-up fashion. Hereinafter, we describe a bottom-up way of building a BPT, i.e. from the leaves up to the root, which corresponds to the historical

way, originally described in [10]. The set of leaves $\mathcal{L}$ is straightforwardly defined from the elements of Ω. The construction of $\mathfrak{T}$ consists of progressively merging, by union, these elements in order to define nodes of increasing size. In practice, the merging order of couples of nodes is related to a chosen merging criterion, based on a given metric on nodes.

A supplementary constraint is given, in practice, when building a BPT. The merging operation has to occur between adjacent nodes. This is why a BPT is built on a graph $(\Omega, \frown)$, and not only on a set Ω. In particular, the set of leaves $\mathcal{L}$ is endowed with an adjacency relation $\frown_{\mathcal{L}}$ trivially inherited from $\frown$, leading to a graph $(\mathcal{L}, \frown_{\mathcal{L}})$ isomorphic to $(\Omega, \frown)$.

The BPT describes the progressive collapsing of $(\mathcal{L}, \frown_{\mathcal{L}})$ onto the trivial graph $(\{\Omega\}, \emptyset)$. This process consists of defining a sequence $(\mathfrak{G}_i = (\mathcal{N}_i, \frown_{\mathcal{N}_i}))_{i=0}^{n}$ (with $n = |\mathcal{L}| - 1$) as follows. First, we set $\mathfrak{G}_0 = (\mathcal{L}, \frown_{\mathcal{L}})$. Then, for each i from 1 to n, we *choose*—according to a given metric—two nodes N_1 and N_2 of $\mathfrak{G}_{i-1}$ such that $N_1 \frown_{\mathcal{N}_{i-1}} N_2$, and we define $\mathfrak{G}_i$ such that $\mathcal{N}_i = (\mathcal{N}_{i-1} \setminus \{N_1, N_2\}) \cup \{N\}$, with $N = N_1 \cup N_2$. In other words, we replace these two nodes by their union. The adjacency $\frown_{\mathcal{N}_i}$ is defined accordingly from $\frown_{\mathcal{N}_{i-1}}$: we remove the edge (N_1, N_2), and we replace each edge (N_1, N_3) and/or (N_2, N_3) by an edge (N, N_3).

The BPT $\mathfrak{T}$ is built in parallel to this progressive collapsing from $\mathfrak{G}_0$ to $\mathfrak{G}_n$. We define a sequence $(\mathfrak{T}_i)_{i=0}^{n}$ as follows. We set $\mathfrak{T}_0 = (\mathcal{N}_0, \emptyset) = (\mathcal{L}, \emptyset)$. Then, for each i from 1 to n, we build $\mathfrak{T}_i$ from $\mathfrak{T}_{i-1}$ by adding the new node $N = N_1 \cup N_2$, and the two edges $N \searrow N_1$ and $N \searrow N_2$. The BPT $\mathfrak{T}$ is finally defined as $\mathfrak{T}_n$.

3. Problem Statement

3.1. Tree Cuts, Partitions and Segmentation

Defining a *cut* within a BPT (and, more generally, within a tree) consists of defining a subset of nodes which are all independent with respect to the hierarchical organization of the tree. In other words, two nodes $N, N' \in \mathcal{N}$ such that $N = N_0 \searrow N_1 \searrow \ldots \searrow N_k = N'$ cannot both belong to a same cut. Trivial examples of cuts of a BPT $\mathfrak{T}$ are the singleton cut $\mathcal{R} = \{N_\Omega\}$ formed by the only root, and the cut $\mathcal{L} = \{N_x\}_{x \in \Omega}$ formed by all the leaves.

A cut is said *total* if it is maximal for the inclusion relation, i.e. if we cannot add a new node hierarchically independent from all the nodes of the cut. For instance, $\mathcal{R}$ and $\mathcal{L}$ are total cuts. A total cut of $\mathfrak{T}$ is indeed a partition of Ω. From a segmentation point of view, a total cut allows one to subdivide the whole image support into elementary regions, generally assumed to be homogeneous

(e.g. spectrally or semantically). In other words, a total cut provides a global segmentation of the scene visualized in the image. This can be useful e.g. for scene labelling in computer vision or for classification purpose in remote sensing.

A cut is said *partial* if it is not necessarily maximal for the inclusion relation. In other words, a partial cut (or simply, a cut) is any subset of a total cut. Equivalently, a partial cut of $\mathfrak{T}$ is a partition of a subset of Ω. From a segmentation point of view, a partial cut allows one to discriminate a region of the image composed by the union of all the nodes inside it, namely the object (or the set of objects, or foreground) from the remainder of the image support, namely the background. In other words, a (partial) cut provides a binary (object vs. background) segmentation of the image. This can be useful e.g. for object detection or image analysis where only a part of the scene is semantically relevant.

The equivalence between cuts and partitions allows for the use of BPTs as a data structure for image segmentation. In the sequel, we will focus on the case of (partial) cuts and their use for object segmentation.

3.2. BPTs as Research Spaces for Segmentation

Based on the above remarks, a BPT constitutes a relevant data structure for image segmentation. In particular, as any of its (partial) cuts provides an object segmentation, it constitutes a research space containing potential segmentation results associated to a given image.

The main virtues of a BPT are its low size and its hierarchical structure. First, it contains $2|\mathcal{L}| - 1$ nodes, with $|\mathcal{L}| = |\Omega|$; in other words, the size of $\mathfrak{T}$ is linear with respect to the image size, i.e. $|\mathcal{N}| = \Theta(|\Omega|)$. Second, the BPT is a (binary) tree structure. This organization of nodes enables the development of efficient "divide-and-conquer" strategies [37], that can lead to low time cost algorithms.

At first sight, BPTs then appear as very good data structures for image segmentation, allowing us to obtain (partial) cuts / object segmentations in a fast and efficient way. Actually, BPTs are indeed very good data structures for such purpose, *provided they are correctly built*. This notion of BPT quality is directly related to the way the BPT is built, given a particular metric impacting the node merging order and for a particular application.

Indeed, BPTs are low-size data structures, that give access to possible partitions of Ω. However, the space of partitions of a set Ω is huge, and grows at a super-exponential ($|\Omega|^{|\Omega|}$) rate with respect to its size. As a direct conse-

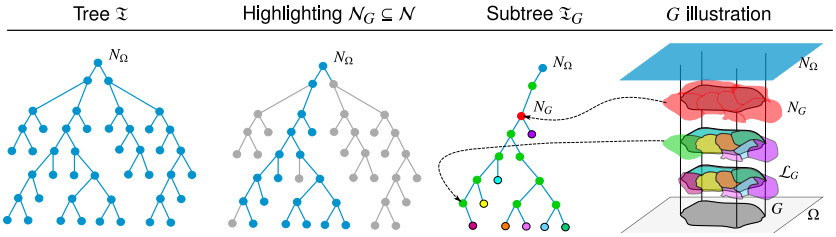

Figure 1: Tree reduction (see Section 4.1). From left to right: a whole tree $\mathfrak{T}$; the subset $\mathcal{N}_G$ of nodes of $\mathcal{N}$ (in blue); the subtree $\mathfrak{T}_G$ composed of the nodes of $\mathcal{N}_G$; the spatial embedding of the nodes $\mathcal{N}_G$ with respect to the ground-truth example G.

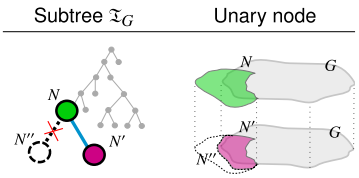

Figure 2: Example of a unary node N with respect to a ground-truth G (see Section 4.1). We have $N \searrow N'$ and $N \cap G = N' \cap G$, whereas $N' \subset N$. The only other node $N'' \subset N$ such that $N \searrow N''$ does not intersect G.

quence, the quality of $\mathfrak{T}$ has a direct impact on the efficiency of any subsequent algorithm that will use it for segmentation purpose.

Being able to evaluate the quality of a BPT built on an image is then of paramount importance.

4. Intrinsic Quality Analysis of a Binary Partition Tree

Assessing the quality of a BPT is equivalent to assessing the relevance of its construction, i.e. the relevance of the features used for successively merging the nodes from the leaves up to the root. In the context of object segmentation, the BPT quality corresponds to its ability to allow for the segmentation of an object of interest exemplified by a ground-truth defined as a subset G of Ω.

A first quality analysis can be carried out by directly observing the BPT with respect to the ground-truth. Indeed, the BPT provides, by its inner structure and its spatial embedding, information about its relevance and its ability to allow for the segmentation of objects similar to this ground-truth.

4.1. Tree Reduction Based on the Ground-Truth

Our purpose, in this first quality analysis, is to evaluate to what extent a BPT $\mathfrak{T}$ defined on $(\Omega, \frown)$ provides nodes matching with a given ground-truth

example $G \subseteq \Omega$. The set of nodes $\mathcal{N}$ of $\mathfrak{T}$ can be subdivided into two families: those that intersect G, and those that do not. The second family is useless here for this analysis. Then, we focus only on the nodes $N \in \mathcal{N}$ intersecting G.

The subset of such nodes can be easily computed in a bottom-up fashion, by first selecting all the leaves $L \in \mathcal{L}$ that intersect G and then preserving iteratively all the parent nodes connected to any such leaves by the $\searrow$ relation, until the root N_Ω, that necessarily satisfies this property (see Figure 1).

The obtained subset of nodes of $\mathcal{N}$, noted $\mathcal{N}_G$, induces a subtree $\mathfrak{T}_G$ of $\mathfrak{T}$, of root N_Ω and with a set of leaves $\mathcal{L}_G$ that correspond to the leaves of $\mathcal{L}$ which intersect G. By construction, this subtree $\mathfrak{T}_G$ *may not be binary*. This means that there may exist nodes N of $\mathcal{N}_G$ that have exactly one child node in $\mathcal{N}_G$. This happens when the only other child node of N does not intersect G (see Figure 2). Such nodes N are called *unary* nodes, by contrast with the *binary* nodes that still have their two children nodes in $\mathcal{N}_G$.

An important property of a unary node N is that its intersection with G is exactly the same as for its only child N' in $\mathcal{N}_G$. In other words, for a unary node N, we have $N \cap G = N' \cap G$, whereas $N' \subset N$. This means that a unary node increases the amount of false positive material with respect to G, compared to its descendants.

In particular, this is true in the upper part of the tree $\mathfrak{T}_G$, between the root N_Ω and the first binary node N_G of $\mathfrak{T}_G$. Indeed, except in the case where N_Ω is itself a binary node in $\mathcal{N}_G$, there exists, within $\mathcal{N}_G$, a sequence of successive nodes $N_\Omega = N_0 \searrow N_1 \searrow \ldots \searrow N_k = N_G$ ($k \geq 1$) such that all N_i ($0 \leq i \leq k-1$) are unary. In such case, we can relevantly remove from $\mathfrak{T}_G$ all these nodes N_i to finally preserve as root the first binary node N_G (see Figure 1). By construction, N_G is the smallest node of $\mathcal{N}$ that includes G.

In order to reduce the space cost of $\mathfrak{T}_G$ in a lossless way, it is also possible to remove the children of the nodes N that are included in G. Indeed, such nodes have their children also included in G. The only useful information to be stored in N is the number ℓ of leaves of the subtree of root N that also gives access to the number of nodes implicitly modeled by N, namely $2\ell - 2$.

The final subtree (still noted $\mathfrak{T}_G$, by abuse of notation) contains all the nodes of interest of $\mathfrak{T}$, with respect to the ground-truth G. Its combinatorial analysis then allows us to obtain first information about the quality of the BPT with respect to the provided ground-truth.

4.2. Relevance of the Quality Analysis

The analysis of the (sub)tree $\mathfrak{T}_G$ provides information about the relevance—with respect to the ground-truth example G—of the way of choosing successively the couples of nodes to merge.

The initial set of leaves is often chosen as $\mathcal{L} = \{\{x\}\}_{x\in\Omega}$, which means that each leaf corresponds to exactly one point of the image. For practical reasons, $\mathcal{L}$ can sometimes be defined as a set of larger nodes (e.g. flat zones or superpixels). The choice of this initial partition is generally non-correlated to the node-merging strategy. Nevertheless, it has an important impact on the relevance of the proposed evaluation framework. In particular, two properties of this initial partition are crucial.

Granularity. The granularity γ derives from the ratio between the size of G (i.e. its number of points) and the size of $\mathcal{L}_G$ (i.e. the number of leaves that intersect G). It is defined as:

$$\gamma = \left(1 + \log_{10} \frac{|G|}{|\mathcal{L}_G|}\right)^{-1} \tag{1}$$

and lies in $(0, 1]$. *The higher the granularity γ, the more relevant the intrinsic quality analysis carried out on $\mathfrak{T}_G$.*

To illustrate this fact, we consider two extremal examples. Let us suppose that $\gamma = 1$. This means that the number of leaves that intersect G is equal to the number of points of G. Then, each leaf contains exactly one point of G, and the ability of the BPT to allow for the definition of a cut that correctly fits G highly depends on the way to progressively merge the nodes. By contrast, let us suppose that $\gamma = 1/(1 + \log_{10}|G|) \simeq 0$ (i.e. its minimal value). This means that one leaf already includes G, thus $\mathfrak{T}_G$ is a trivial tree composed of one node. Then, the initial partition already determines the adequacy between the BPT and G; the way of building the tree has no actual influence. More generally, the proposed logarithmic formulation of granularity (which differs from the one initially proposed in [35, (6)]), allows to better model the difference of order of magnitude between the size of the ground-truth and the number of leaves that intersect it. Indeed, a granularity $\gamma = \frac{1}{k}$ corresponds to a difference of $k - 1$ orders of magnitude, i.e. a ratio of 10^{k-1}.

Discordance. The discordance δ is defined as the relative quantitative error on the size of G induced by $\mathcal{L}_G$, and more precisely by the nodes that partially

intersect G. It is defined as:

$$\delta = \frac{1}{|G|} \sum_{L \in \mathcal{L}_G} \min\{|L \setminus G|, |L \cap G|\} \tag{2}$$

and lies in $[0, 1]$. *The lower the discordance, the more relevant the intrinsic quality analysis carried out on* $\mathfrak{T}_G$. To illustrate this fact, let us consider two extremal examples. Let us suppose that $\delta = 0$. This means that the initial partition provides a set of leaves that perfectly fits G. Then, the quality of the BPT directly depends on the ability to progressively merge these nodes in order to retrieve G. By contrast, let us assume that $\delta = 1$. This means that many leaves of $\mathcal{L}$ (and actually all the leaves of $\mathcal{L}_G$) intersect both G and the remainder of Ω. Then, the ability to finally obtain a good segmentation of G is quite low and weakly depends on the ability to merge the nodes when building the BPT.

Both granularity and discordance have to provide satisfactory values if one wants to relevantly assess the structural quality of a BPT. For G sufficiently large, a high value of granularity will imply a low discordance. However, the counterpart is not true: a low discordance can be obtained for a low value of granularity (e.g. if a leaf L already fits G). As a consequence, it is important to carefully compute these two scores before any such study.

Of course, such precaution becomes useless if the initial partition $\mathcal{L}$ is isomorphic with Ω, i.e. when the initial partition is composed of leaves that correspond to points of the image. Indeed, in such case, we have $\gamma = 1$ and $\delta = 0$: the quality of the final BPT only depends on the node merging process, since the latter is not biased by the initial grouping of points into primitive regions.

Remark. Various metrics have been defined in the literature for assessing the quality of a partition with respect to a ground-truth, essentially for evaluating superpixel paradigms [38]. Two of these metrics, dedicated to measure under-segmentation errors, are close to the discordance δ, namely the opposite of the Achievable Segmentation Accuracy (ASA) defined in [39] and the undersegmentation error UE defined in [40]. These metrics have been designed for assessing the similarity between two partitions, with the underlying assumptions that they both present comparable properties. Our hypotheses are somewhat different. Indeed, we assess the adequacy of the initial set of leaves $\mathcal{L}$, namely a partition of Ω, with a ground-truth G which is a subset of Ω. However, G may also be implicitly seen as a (binary) partition of Ω, namely $\Pi_G = \{G, \Omega \setminus G\}$.

In this context, we can rewrite the opposite of ASA [38, (3)] and UE [38, (4)] under our hypotheses, leading to the two following formulae:

$$1 - ASA(\Pi_G, \mathcal{L}) = \frac{1}{|\Omega|} \sum_{L \in \mathcal{L}_G} \min\{|L \setminus G|, |L \cap G|\} = \frac{|G|}{|\Omega|}\delta \tag{3}$$

$$UE(\Pi_G, \mathcal{L}) = \frac{2}{|\Omega|} \sum_{L \in \mathcal{L}_G} \min\{|L \setminus G|, |L \cap G|\} = \frac{2|G|}{|\Omega|}\delta \tag{4}$$

In other words, under our specific hypotheses, the both metrics $1 - ASA$ and UE (which are the same here, up to a multiplicative factor) differ from the discordance δ by the fact that δ normalizes the pointwise measured error by the size of the ground-truth G. By contrast, the normalization carried out in (3–4) relies on the size $|\Omega|$ of the image support, since these metrics are natively defined for comparing two partitions with same properties, and in particular with comparable cardinalities. In our case, it is of course relevant to normalize with respect to $|G|$ instead of $|\Omega|$ since the relative error should depend on the considered ground-truth, and not of the overall size of the image. Concerning the granularity γ, we did not find similar metrics in the literature. Indeed, the purpose is here to estimate how much a partition subdivides a ground-truth object. As a consequence, such metric cannot be compared to metrics which purpose is to assess the correspondence between partitions with comparable cardinalities.

4.3. Intrinsic Quality Analysis

From now on, we assume that both the γ and δ scores are sufficiently good to carry out an intrinsic quality analysis. In other words, we assume that the partition of the leaves of the BPT $\mathfrak{T}$ is sufficiently fine, and globally well-fitting the ground-truth example G. (Otherwise, it is plain that the user has to reconsider his/her way to define the initial partition, in order to adapt it to the targeted objects.)

4.3.1. Combinatorial Analysis

The (sub)tree $\mathfrak{T}_G$ is composed of n nodes (including one root and ℓ leaves) and $n - 1$ edges (defined from the $\searrow$ relation between parent-child nodes). Observing the status of these nodes and carrying out a combinatorial analysis on the different populations provide us with quality clues of the BPT.

Let us first introduce the following terminology. A node N of $\mathcal{N}_G$ is *pure* (resp. *impure*) if $N \subseteq G$ (resp. $N \not\subseteq G$).

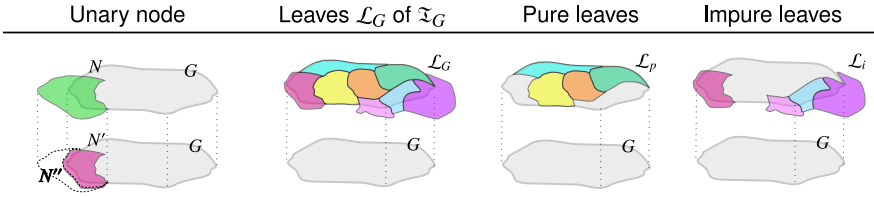

Figure 3: From left to right: a unary node N (impure) composed of one (impure) node N' and one node N'' out of G; the set $\mathcal{L}_G$ of leaves intersecting G; the subset of pure leaves of $\mathcal{L}_G$; the subset of impure leaves of $\mathcal{L}_G$.

The classification of the leaves of $\mathcal{L}_G$ into pure / impure can be made easily by observing their support, compared to G (see Figure 3). We set ℓ_p and ℓ_i the number of pure and impure leaves, respectively. In particular, we have $\ell_p + \ell_i = \ell$, with ℓ of the same order as $|G|$, due to high granularity hypothesis.

The purity / impurity of the other nodes can be determined easily, by considering that:

- if N is a unary node, then it is impure;
- if N is a binary node, with $N \searrow N', N''$, then
  - if N' or N'' is impure, then N is impure;
  - if N' and N'' are pure, then N is pure.

A good BPT construction, i.e. leading to a BPT that may allow to carry out an accurate segmentation of a target object, should preserve as much as possible the purity of nodes, and avoid (1) to merge pure and impure nodes, and (2) to increase the size of impure nodes. Based on these assertions:

- the merging leading to a binary node from two pure nodes in $\mathfrak{T}_G$ is a good operation, as it allows one to converge towards G;
- the merging leading to a binary node from two impure nodes in $\mathfrak{T}_G$ is a neutral operation; it avoids to deteriorate pure areas and to increase the amount of points out of G;
- the merging leading to a binary node from a pure and an impure node in $\mathfrak{T}_G$ is a bad operation, as it makes the result diverging from G;
- the merging leading to a unary node in $\mathfrak{T}_G$ is a bad operation, as it creates from an impure (and sometimes pure) node another impure node with a greater amount of points out of G.

By simply observing the subtree $\mathfrak{T}_G$, it is possible to count the number of each kind of nodes: u_i, b_{pp}, b_{ii} and b_{pi}, for the unary nodes, and binary nodes built from pure–pure, impure–impure, pure–impure couples, respectively. We have $u_i + b_{pp} + b_{ii} + b_{pi} + \ell_p + \ell_i = n$, and a good BPT should minimize u_i and b_{pi}, whereas maximizing b_{pp} and b_{ii}. In particular, a perfect BPT should satisfy:

$$\begin{cases} u_i &= 0 \\ b_{pp} &= \ell_p - 1 \\ b_{ii} &= \ell_i - 1 \\ b_{pi} &= 1 \end{cases}$$

(except when $\ell_i = 0$, where b_{pp} should be equal to $\ell - 1$, and all others to 0).

From this classification of nodes, and the combinatorial analysis of their population, it is then possible to build a wide range of structural measures that quantify the difference of quality between BPTs. Two examples of such structural measures are:

$$C_1 = \frac{b_{pp} + b_{ii}}{u_i + b_{pp} + b_{ii} + b_{pi}} \tag{5}$$

$$C_2 = \frac{b_{pp}}{b_{pp} + b_{pi}} \tag{6}$$

which take their values in $[0, 1]$. The higher the value, the better the BPT. In particular, C_1 evaluates the ability of a BPT to avoid mixing pure and impure nodes whereas C_2 evaluates its ability to preserve nodes pure.

4.3.2. Quantitative Analysis

A more quantitative assessment of the quality of a BPT can also be carried out by observing the lowest set including G and the greatest set included in G which can be built from $\mathfrak{T}_G$.

Let us first focus on the lowest set including G. By construction, this set is indeed the root of $\mathfrak{T}_G$, namely N_G. The interesting information carried by N_G is the amount of points outside G. More precisely, this amount $|N_G \setminus G| = |N_G| - |G|$ has to be compared to the amount that could be theoretically obtained from the initial partition of the leaves $\mathcal{L}_G$, namely $|\bigcup \mathcal{L}_G \setminus G| = |\bigcup \mathcal{L}_G| - |G|$. Computing the difference or the ratio between these two values, namely:

$$Q_1 = |N_G| - |\bigcup \mathcal{L}_G| \tag{7}$$

$$Q_2 = \frac{|N_G| - |G|}{|\bigcup \mathcal{L}_G| - |G|} \tag{8}$$

(possibly normalized by the size of G) allows us to assess the quantitative error related to the existence of unary nodes in $\mathfrak{T}_G$, i.e. the addition of non-relevant zones to the expected exhaustive segmentation of G. The lower these values, the better the ability of the BPT to use at best the adequacy of the initial partition to the ground-truth. It is worth mentioning that these metrics are relevant when the object G is small compared to the image (e.g. in remote sensing images); indeed, when G is large (e.g. in natural images) it often happens that N_G is very close (or even equal) to G, and in such case, Q_1 and Q_2 do not provide useful information on the quality of the BPTs.

Second, let us focus on the greatest set included in G. It is defined as the union of all the pure nodes of $\mathcal{N}_G$ whose parents are impure; we note this set $\mathcal{N}_p$. A quality metric can be defined as the ratio between this number $|\mathcal{N}_p|$ of pure nodes with impure parents, and the number of the (pure) leaves of the BPT that lie into G; we note $\mathcal{L}_p$ this set of leaves. The quality metric is then defined as:

$$Q_3 = \frac{|\mathcal{N}_p|}{|\mathcal{L}_p|} \tag{9}$$

and it takes its values in $(0, 1]$. The lower this metric, the better the BPT. Indeed, $Q_3 = 1$ means that all the leaves of $\mathcal{L}_p$ are also nodes of $\mathcal{N}_p$; in other words, it was not possible to merge any two pure leaves to build a pure node. By contrast, $Q_3 \simeq 0$ means that the number of nodes of $\mathcal{N}_p$ is very low compared to the number of pure leaves of $\mathcal{L}_p$; in other words, nearly all these pure leaves could be merged together whereas preserving the purity of the induced nodes.

5. Extrinsic Quality Analysis of a Binary Partition Tree

Beyond the intrinsic quality criteria provided by its hierarchical structure, the relevance of a BPT is also related to its ability to provide good segmentation results with respect to the user's point of view. In this context, evaluating the quality of a BPT requires to define a dedicated quality metric, namely a function Λ, which associates a value $\Lambda(S) \in \mathbb{R}$ to each potential object segmentation $S \subseteq \Omega$. This metric Λ is determined by the user and depends on his/her expectations regarding the properties of the result S. Most of the time, such quality metric Λ depends on a ground-truth $G \subseteq \Omega$, that corresponds to the perfect expected segmentation; in such cases it is denoted as Λ_G.

The core idea is to find a (partial) cut $\mathcal{C} \subseteq \mathcal{N}$ within the BPT $\mathfrak{T}$ forming a segmented region $S(\mathcal{C}) = \bigcup \mathcal{C} \subseteq \Omega$ that optimizes the quality metric Λ, i.e. to

solve the following optimization problem:

$$\widehat{\mathcal{C}} = \arg\max_{\mathcal{C}\subseteq\mathcal{N}} \Lambda(S(\mathcal{C})) \tag{10}$$

This equation would remain valid even if $\mathcal{C}$ was not a cut but only a subset of $\mathcal{N}$, since the definition of $S(\mathcal{C})$ from $\mathcal{C}$ is made by union, and then hides the effects of overlapping between nodes. However it is indeed sufficient to consider the actual (partial) cuts within $\mathcal{N}$, since for any segmentation induced by a subset of $\mathcal{N}$, there exists a (partial) cut $\mathcal{C}$ such that $S(\mathcal{C})$ is this segmentation.

More generally, there may exist many (partial) cuts that lead to a same segmentation. This derives from the fact that if two nodes N_1, N_2 belong to a cut whereas $N = N_1 \cup N_2$ is also a node (i.e. N is the parent node of N_1, N_2), then N_1 and N_2 can be replaced by N within the cut without modifying the induced segmentation. In practice, for a given segmentation, i.e. for a given set of cuts leading to a same segmentation, we will always consider the cut of lower size, i.e. the cut which is the closest from the root. This choice is motivated by the fact that the smaller the cut, the better the ability of the BPT to allow for the chosen segmentation (the best scenario being a segmentation directly given by a unique node, i.e. a singleton cut).

Since we consider exactly one cut per potential segmentation, namely the cut involving the least number of nodes, there is a bijection between these cuts $\mathcal{C}$ and the induced segmentations $S(\mathcal{C})$. In this context, for the sake of concision, we will note $\Lambda(\mathcal{C})$ instead of $\Lambda(S(\mathcal{C}))$.

Under these hypotheses, the total number of cuts to be consisered for solving the optimization problem (10) is $2^{|\mathcal{L}|}$ with[1] $|\mathcal{L}| \leq |\Omega|$ (vs. $2^{2|\mathcal{L}|-1}$ if we had considered all the subsets of $\mathcal{N}$).

5.1. Space Reduction

The optimization problem (10) admits a solution, since the number of considered cuts is finite, but the space of potential solutions is exponentially high. In order to reduce the size of this problem, some hypotheses can be made, in particular when the metric Λ relies on a ground-truth G.

On the one hand, one can assume that a node that is fully outside G does not contribute by its own to the solution $\widehat{\mathcal{C}}$. In other words, such node may be potentially relevant only if its parent node intersects G. By contraposition, if

[1] $|\mathcal{L}| = |\Omega|$ if we consider that each leaf is composed of one point of Ω, and $|\mathcal{L}| < |\Omega|$ if we define the leaves from a partition of Ω obtained e.g. by a superpixel preprocessing.

a node does not intersect G, one can remove its two children nodes and their induced subtrees. This is equivalent to preserve the smallest binary subtree of the BPT that contains all the nodes that intersect G. The only remaining nodes of this subtree that do not intersect G are then those required to preserve this subtree binary.

On the other hand, if a node is fully inside G, one can also assume that it will contribute to the definition of the optimal segmentation more relevantly than any of its strict subsets. Consequently, one can remove its two children nodes and their induced subtrees.

As in Section 4.1, it is also possible to choose as root for this binary subtree the node N_G, namely the smallest node of $\mathcal{N}$ that includes G.

The binary subtree finally obtained provides a reduced subset of nodes (noted $\mathcal{N}_G$ from now on) and in particular of leaves (noted $\mathcal{L}_G$ from now on), that can be relevantly used for solving (10). This number of leaves $|\mathcal{L}_G|$ is lower or equal to $|N_G|$, with in general $|N_G| \ll |\mathcal{L}|$, and the number of cuts to be considered is then $2^{|\mathcal{L}_G|} \ll 2^{|N_G|} \ll 2^{|\mathcal{L}|}$.

5.2. Partial Cut Construction

In order to solve the optimization problem (10), it is indeed required to compute the value of Λ for the segmentation associated to each candidate cut within the considered (sub)tree. Based on the above discussion, there exist exactly $2^{|\mathcal{L}_G|}$ cuts leading each to a distinct segmentation. A simple way of defining these $2^{|\mathcal{L}_G|}$ segmentations is to consider the power set of $\mathcal{L}_G$. Indeed, each subset of leaves corresponds to exactly one cut $\mathcal{C}$ and one segmentation $S(\mathcal{C})$. In other words, generating the exhaustive set of segmentations / cuts can be carried out by a simple scanning of the power-set of $\mathcal{L}_G$.

However, a cut defined as a subset of leaves of $\mathcal{L}_G$ is not necessarily a valid cut for our criteria, since there may exist another cut of lower size, leading to the same segmentation, as discussed above. For instance, the cut $\mathcal{L}_G$ (corresponding to the whole set of leaves) leads to a segmentation result which is equal to N_G, whereas the cut $\{N_G\}$ (composed by a single node, namely the root of $\mathfrak{T}_G$) leads to the same segmentation; in this example, the "good" cut is $\{N_G\}$ and not $\mathcal{L}_G$. In order to identify the cuts of lowest size leading to the segmentations of interest, it is possible to rely on a simple recursive approach. In particular, for a (sub)tree $\mathfrak{T}_N$ with a root N, two cases can occur: (1) if N is the only node

of the tree, namely a leaf, then the set of correct cuts $\Pi(N)$ is defined as:

$$\Pi(N) = \{\emptyset, \{N\}\} \tag{11}$$

whereas (2) if N has two children nodes N_1 and N_2, then the set of correct cuts $\Pi(N)$ is (recursively) defined as:

$$\Pi(N) = \{U(p) \mid p \in \Pi(N_1) \times \Pi(N_2)\} \cup \{\{N\}\} \setminus \{\{N_1, N_2\}\} \tag{12}$$

where U is the binary union function defined as $U((X, Y)) = X \cup Y$. (In (12), the minimality of the cuts is guaranteed by the "$\cup\{\{N\}\} \setminus \{\{N_1, N_2\}\}$" operation, which substitutes $\{\{N\}\}$ to $\{\{N_1, N_2\}\}$, since $N = N_1 \cup N_2$.) The set of the correct cuts in $\mathfrak{T}_G$ is then defined as $\Pi(N_G)$. It has a size $2^{|\mathcal{L}_G|}$, and can be computed in a time $\mathcal{O}(2^{|\mathcal{L}_G|})$ if the tree is well-balanced.

Despite the potentially low size of the set of leaves $\mathcal{L}_G$ (compared to $\mathcal{L}$ and Ω), the time and space complexity for exhaustively exploring the whole set of candidate cuts for solving (10) remains exponential. In the next sections, we investigate exact and approximated ways of solving (10) under certain constraints / hypotheses, and thus assessing the quality of a BPT with respect to Λ.

5.3. Exact Problem Solving: The Case of Small Cuts

In first approximation, one may assume that the best segmentation result with respect to Λ may be obtained from exactly one node of the tree. In other words, this would mean that there exists one node N within the nodes of $\mathcal{N}_G$ such that the cut defined as the singleton $\{N\}$ leads to a segmentation $S(\{N\}) = N$ that maximizes Λ, and is then a solution of (10). Following this hypothesis of one-node optimal cut, solving (10) would only require to test each node of the subtree $\mathfrak{T}_{N_G}$. Since these nodes are already available within the subtree, the computational cost of such approach is $\lambda|\mathcal{N}_G|$, where λ is the time cost for computing Λ on a given segmentation.

However, it is not frequent that an optimal cut be composed of exactly one node. In practice, for obtaining a correct segmentation from a BPT, it is generally required to consider a cut with (not too) many nodes. If this number p of nodes and the size of $\mathcal{N}_G$ are both small enough, then (10) can be solved exactly—under the additional constraint that the sought optimal cut has a size lower than p—in a time $\lambda \sum_{k=0}^{p} \binom{|\mathcal{N}_G|}{k} = \mathcal{O}(\lambda|\mathcal{N}_G|^p)$, by generating and testing the small cuts from the subtree $\mathfrak{T}_G$. In practice, these small cuts can be simply obtained from (11–12) by removing all the cuts of size strictly greater than p.

This can be done, for instance, by setting $U(X,Y) = X \cup Y$ if $|X \cup Y| \leq p$ and $\emptyset$ otherwise. In other words a polynomial time cost is required for the exact solving of (10) for small cuts, which is tractable for size of cuts sufficiently low.

5.4. Exact Problem Solving: The Case of Separable Optimal Cuts

In the most favourable cases, the chosen quality metric Λ induces a property of separability (or closely related properties) on the optimal cuts. Previous works dedicated to such hypotheses may be found e.g. in [37, 41]. This means that the optimal cut / segmentation with respect to Λ within a tree of root N is defined as the union of the two optimal cuts / segmentations within the two subtrees which roots are the two chidren nodes N_1 and N_2 of N. (Examples of such metrics Λ_G are the amount of true positives $|S \cap G|$, the (opposite) of the amount of false positives $-|S \setminus G|$, or positive linear combinations of both.)

In such case, the optimization problem (10) can be solved by setting:

$$\widehat{\mathcal{C}}(N) = \arg\max\{\Lambda(\emptyset), \Lambda(\{N\})\} \tag{13}$$

if N has no chidren nodes, and:

$$\widehat{\mathcal{C}}(N) = \begin{cases} \{N\} & \text{if } \widehat{\mathcal{C}}(N_1) = \{N_1\} \text{ and } \widehat{\mathcal{C}}(N_2) = \{N_2\} \\ \widehat{\mathcal{C}}(N_1) \cup \widehat{\mathcal{C}}(N_2) & \text{otherwise} \end{cases} \tag{14}$$

if N has two chidren nodes N_1 and N_2. The solution of (10) is then defined as $\widehat{\mathcal{C}}(N_G)$. From this formulation, one can observe that it is sufficient to investigate each leaf independently in order to determine the best segmentation and the associated best cut in a time $\mathcal{O}(\lambda|\mathcal{N}_G|)$.

However, it may happen that the best cut is composed of too many nodes with respect to the user's expectation, as previously observed in a closely related study [30]. Then, inspired by [30], it is possible to constrain (10) to find the best cut being composed of at most p nodes for a given p, or more generally to find the $p+1$ best cuts composed of k nodes, respectively, for any k in $[0, p]$. To this end, it is then required to enrich (13–14), by storing a vector $\mathcal{V} = [\widehat{\mathcal{C}}_i]_{i=0}^{p}$ of cuts $\mathcal{C}_i$ of size i instead of one cut. More precisely, the i-th element ($i \in [0, p]$) of this vector contains the optimal cut of size i (if such cut exists) at the current stage. In particular, (13) then becomes:

$$\widehat{\mathcal{V}}(N) = [\emptyset, \{N\}, nil, \ldots, nil]$$

where *nil* means that there exists no optimal cut of the corresponding size;

whereas (14) becomes:

$$
\begin{aligned}
&\widehat{\mathcal{C}_0}(N) = \emptyset \\
&\widehat{\mathcal{C}_1}(N) = \left\{ \begin{array}{ll} \arg\max \left\{ \begin{array}{l} \Lambda(\widehat{\mathcal{C}_0}(N_1) \cup \widehat{\mathcal{C}_1}(N_2)) \\ \Lambda(\widehat{\mathcal{C}_1}(N_1) \cup \widehat{\mathcal{C}_0}(N_2)) \\ \Lambda(\{N\}) \end{array} \right\} & \text{if } \left\{ \begin{array}{l} \widehat{\mathcal{C}_1}(N_1) = \{N_1\} \\ \widehat{\mathcal{C}_1}(N_2) = \{N_2\} \end{array} \right. \\ \arg\max \left\{ \begin{array}{l} \Lambda(\widehat{\mathcal{C}_0}(N_1) \cup \widehat{\mathcal{C}_1}(N_2)) \\ \Lambda(\widehat{\mathcal{C}_1}(N_1) \cup \widehat{\mathcal{C}_0}(N_2)) \end{array} \right\} & \text{otherwise} \end{array} \right. \\
&\widehat{\mathcal{C}_2}(N) = \left\{ \begin{array}{ll} \arg\max \left\{ \begin{array}{l} \Lambda(\widehat{\mathcal{C}_0}(N_1) \cup \widehat{\mathcal{C}_2}(N_2)) \\ \Lambda(\widehat{\mathcal{C}_2}(N_1) \cup \widehat{\mathcal{C}_0}(N_2)) \end{array} \right\} & \text{if } \left\{ \begin{array}{l} \widehat{\mathcal{C}_1}(N_1) = \{N_1\} \\ \widehat{\mathcal{C}_1}(N_2) = \{N_2\} \end{array} \right. \\ \arg\max \left\{ \begin{array}{l} \Lambda(\widehat{\mathcal{C}_0}(N_1) \cup \widehat{\mathcal{C}_2}(N_2)) \\ \Lambda(\widehat{\mathcal{C}_1}(N_1) \cup \widehat{\mathcal{C}_1}(N_2)) \\ \Lambda(\widehat{\mathcal{C}_2}(N_1) \cup \widehat{\mathcal{C}_0}(N_2)) \end{array} \right\} & \text{otherwise} \end{array} \right. \\
&\widehat{\mathcal{C}_k}(N) = \arg\max\{\Lambda(\widehat{\mathcal{C}_i}(N_1) \cup \widehat{\mathcal{C}_{k-i}}(N_2))\}_{i=0}^{k} \ \ \forall k \in [3, p]
\end{aligned}
$$

with $nil \cup X = X \cup nil = nil$ for any set X. Based on this formulation, the vector $\mathcal{V}(N_G)$ can be computed in a time $\mathcal{O}(\lambda|\mathcal{N}_G|p^2)$, which remains tractable even for high values of p. This vector $\mathcal{V}(N_G)$ provide the $p+1$ exact solutions to the optimization problem (10), under the constraints that the partial cuts are of sizes $k \in [0, p]$.

5.5. Approximate Problem Solving: The Case of Separable Metrics

We now consider a case of separability more general / less restrictive than the previous one. We assume that the useful information required to compute the value Λ for a given segmentation / cut within a tree of root N can be directly obtained from the information associated to the two induced cuts within the subtrees of roots N_1 and N_2 where N_1, N_2 are the children nodes of N.

Under these hypotheses, we note K the function which provides, for any cut, the information that allows us to compute its value for Λ, and we note Ξ the function that explicitly computes this value from the information. In other words, we have:

$$\Lambda(\mathcal{C}) = \Xi(K(\mathcal{C})) \tag{15}$$

Due to the above separability property, we assume that it is possible to compute the information associated to a cut from the information associated to two cuts which subdivide it. We note Φ the function that allows to carry out this agglomerative computation. As a consequence, we then need to explictly define K for the end-cases, namely $K(\emptyset)$ and $K(\{L\})$ for any $L \in \mathcal{L}_G$, whereas for any

other cut $\mathcal{C}$ we have:

$$K(\mathcal{C}) = \begin{cases} \Phi(K(\{N_1\}), K(\{N_2\})) & \text{if } \mathcal{C} = \{N\} = \{N_1 \cup N_2\} \\ \Phi(K(\mathcal{C}_1), K(\mathcal{C}_2)) & \text{otherwise, with } \mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2 \end{cases} \tag{16}$$

In other words, we need to define Ξ, Φ, $K(\emptyset)$ and $K(\{L\})$.

In order to exemplify these notions, let us consider three metrics, namely the Dice score Λ_G^D, the Jaccard index Λ_G^J and the "true positive minus false positive" metric Λ_G^P (evoked in the previous subsection), all with respect to a ground-truth G. These three metrics, defined as:

$$\Lambda_G^D(S) = \frac{2|S \cap G|}{|S| + |G|} \tag{17}$$

$$\Lambda_G^J(S) = \frac{|S \cap G|}{|S \cup G|} \tag{18}$$

$$\Lambda_G^P(S) = |S \cap G| - |S \setminus G| \tag{19}$$

are indeed separable and each of the $\Lambda_G^\star$ can be defined from (15–16) by considering the following definitions of $\Xi_G^\star$, $\Phi_G^\star$, $K_G^\star(\emptyset)$ and $K_G^\star(\{L\})$:

$$\begin{aligned} K_G^D(\emptyset) = K_G^J(\emptyset) = K_G^P(\emptyset) &= (0, 0) \\ K_G^D(\{L\}) &= (|L \cap G|, |L|) \\ K_G^J(\{L\}) = K_G^P(\{L\}) &= (|L \cap G|, |L \setminus G|) \end{aligned}$$

and:

$$\Phi_G^D = \Phi_G^J = \Phi_G^P(\{L\}) = ((x_1, y_1), (x_2, y_2)) \mapsto (x_1 + x_2, y_1 + y_2)$$

and:

$$\begin{aligned} \Xi_G^D &= (x, y) \mapsto \frac{2x}{y + |G|} \\ \Xi_G^J &= (x, y) \mapsto \frac{x}{y + |G|} \\ \Xi_G^P &= (x, y) \mapsto x - y \end{aligned}$$

Let us now consider one cut $\mathcal{C}_\star$ within the set of the $|2^{\mathcal{L}}|$ considered cuts of the processed subtree $\mathfrak{T}_{N_G}$. As discussed above, this cut corresponds to one of the $|2^{\mathcal{L}_G}|$ possible segmentations which can be obtained from this subtree, and this cut is completely determined by the choice of a subset $\mathcal{L}_\star$ of leaves within

$\mathcal{L}_G$. More precisely, this cut $\mathcal{C}_\star$ induced by $\mathcal{L}_\star$ is defined as $\mathcal{C}_\star = C_{\mathcal{L}_\star}(N_G)$ with:

$$C_{\mathcal{L}_\star}(N) = \{N\} \cap \mathcal{L}_\star$$

if N is a node of the subtree $\mathfrak{T}_{N_G}$ without children nodes, and:

$$C_{\mathcal{L}_\star}(N) = \begin{cases} \{N\} & \text{if } C_{\mathcal{L}_\star}(N_1) = \{N_1\} \text{ and } C_{\mathcal{L}_\star}(N_2) = \{N_2\} \\ C_{\mathcal{L}_\star}(N_1) \cup C_{\mathcal{L}_\star}(N_2) & \text{otherwise} \end{cases}$$

if N has two chidren nodes N_1 and N_2 in $\mathfrak{T}_{N_G}$.

This cut $\mathcal{C}_\star$ can be computed in a time $\mathcal{O}(|\mathcal{L}_G|)$. Based on (15–16) we can compute the value $\Lambda(\mathcal{C}_\star)$ in a time $\mathcal{O}((\varphi + k)|\mathcal{L}_G| + \xi)$ where ξ, k and φ are the time costs of the application of the functions Ξ, Φ and K, respectively.

It is important to note that it is also possible to compute the Λ values for all the intermediate cuts leading from the void or singleton cuts computed at the leaves of the subtree $\mathfrak{T}_{N_G}$, up to its root. In such case, (15) is no longer applied once, but $|\mathcal{N}_G|$ times, and the overall time cost becomes $\mathcal{O}(\xi(\varphi + k)|\mathcal{L}_G|)$. As a counterpart, it becomes possible to compute the Λ value(s) of no longer one but many cuts, with a number that varies between 1 in the worst case and $|\mathcal{N}_G|$ in the best case.

For many metrics, the time cost for the functions Ξ, Φ and K is constant, and in such case, the time cost for computing the Λ value(s) of a cut $\mathcal{C}_\star$ and all the induced cuts within the subtrees of $\mathfrak{T}_{N_G}$ is indeed $\mathcal{O}(|\mathcal{L}_G|)$. This is for instance the case of the three metrics exemplified above.

The overall number of cuts is $2^{|\mathcal{L}_G|}$. As a consequence, due to the exponential size of this space of potential solutions, an exhaustive research is generally intractable. Nevertheless, due to the time cost of the evaluation a single cut and the possibility to compute many results when scanning once the tree, it is possible to carry out a partial research over (more than) n cuts in a time cost $\mathcal{O}(n|\mathcal{L}_G|)$.

In addition, if we store the K information at each node of the subtree $\mathfrak{T}_{N_G}$ when processing it for a given cut $\mathcal{C}_\star$, then it is possible to compute Λ for a cut $\mathcal{C}'_\star$ which differs in exactly one leaf node compared to $\mathcal{C}_\star$ (plus the Λ values for the intermediate cuts) in a time $\mathcal{O}(\log_2 |\mathcal{N}_G|)$ if the tree is well-balanced.

In other words, by combining a random exploration of n cuts and neighbouring explorations of m cuts around each of these n cuts, we can compute the Λ values for (at least) $\mathcal{O}(nm)$ cuts in a time $\mathcal{O}(n|\mathcal{L}_G| + nm \log_2 |\mathcal{L}_G|)$. In particular, the values of n and m can be defined by the user depending on his/her

tolerance to the uncertainty associated to a non-exhaustive exploration of the research space.

6. Experiments

To illustrate our framework and show its usefulness, as well as its versatility, we involve it in a classical segmentation task of natural images from well-known datasets, often considered in computer vision challenges. In particular, we use our framework for assessing the relevance of various features, based on color homogeneity and simple geometric properties, for the construction of BPTs dedicated to allow for the segmentation of semantic objects.

We recall that the purpose of this experimental study is *not* to evaluate the gain of considering BPT-based segmentation procedures vs. other state-of-the-art segmentation algorithms. (In particular, we do not compare BPTs with other segmentation paradigms.) The purpose is to show that the proposed BPT evaluation framework can be used to determine the most relevant BPT construction features among a population of candidate features.

For the sake of research reproducibility, and for making this work actually useful for the community, an implementation of the proposed framework can be found at the following url: `https://github.com/yonmi/TreeEvaluation`.

6.1. Datasets

Three image collections are used in this experimental study. Such datasets are classically considered in computer vision challenges; they contain natural images representing various scenes:

- the Grabcut [42] dataset which is composed of 50 images representing quite complex shapes but where the foreground and background present disjoint color distributions;
- the Weizmann-1-object [43] dataset (called "Weizmann" in the sequel) which contains 100 color images quite similar to Grabcut;
- the Visual Object Classes Challenge 2012, segmentation task (VOC2012-segmentation) [44] (called "VOC" in the sequel) from which we consider an illustrative subset of 26 images[2]. This dataset has simpler shapes than Grabcut and Weizmann but more complex appearances, where background and foreground color distributions may sometimes overlap.

[2] `http://host.robots.ox.ac.uk/pascal/VOC/voc2012/segexamples/`

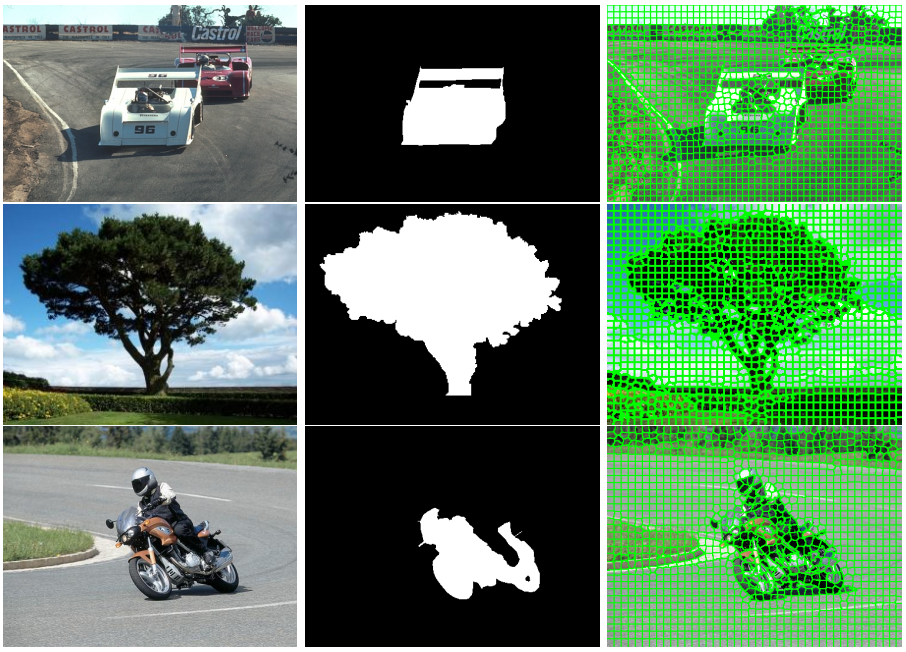

Figure 4: Examples of color images from the three considered datasets (left), with their associated ground-truths (middle; see Section 6.1) and their associated superpixel partition (right; see Section 6.2) Top: Grabcut. Center: Weizmann. Bottom: VOC.

All the images from these three datasets are endowed with ground-truths. For both Grabcut and Weizmann, each ground-truth image is binary and delineates one single object of interest whereas in VOC, different objects of interest can be observable in the ground-truth image, each one associated with a unique label value (in such case, we choose one of these objects). Examples of color images and their associated ground-truths are presented in Figure 4.

In the sequel, a color image $I : \Omega \to \mathbb{N}^3$ associates to each point $x \in \Omega$ a RGB triple of spectral intensities: $I(x) = (I_R(x), I_G(x), I_B(x))$. In our specific case, Ω is a matrix of pixels, namely a subset of the Cartesian grid.

6.2. Initial Partitioning

As discussed in Section 4.2, a BPT is built from a set of leaves which can be either the pixels of the image or some regions obtained from a preprocessing step, for instance a superpixel partitioning. In our experiments, we consider such superpixel partitioning and we rely on the well-known SLIC method [45]. SLIC is a classical superpixel approach, that builds image segments using a k-means clustering in the color-(x,y) space. Despite its simplicity, SLIC adheres well to object boundaries and is recognized as faster and more memory efficient

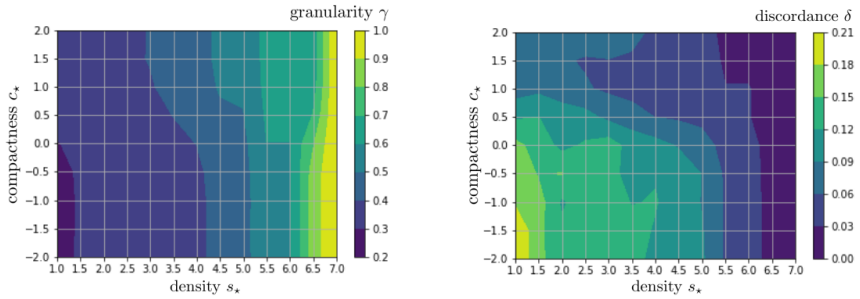

Figure 5: Granularity γ (left) and discordance δ (right) mean values for the tested images of the datasets Grabcut, Weizmann and VOC, depending on the density (x-axis) and the compacity (y-axis). The results are expressed in log-space ($\log_{10}$ for compacity and $\log_2$ up to a multiplicative factor of 250 for the size of the partition, which derives from the compacity).

than most of the state-of-the-art approaches for generating superpixels.

The main two parameters of SLIC are the compactness of the superpixels, noted $c_\star$, that trades off color similarity and spatial proximity, and the number of centers for k-means, noted $s_\star$, that defines the number of superpixels within the computed partition. For our experiments, we fix these parameters in order to define superpixel partitions of the images of each of the three considered datasets.

These parameters are defined with respect to the two measures of granularity γ (1) and discordance δ (2) given in Section 4.2. For each image, we compute a SLIC partition with parameters $c_\star$ within the set of values $[10^{-2}, 10^2]$ and $s_\star$ within the set of values $[250 \cdot 2^1, 250 \cdot 2^7]$. For couples of values $(c_\star, s_\star)$ sampled in the research space, we then compute the average values of the two measures γ and δ over all the images. From these results, illustrated in Figure 5, we empirically choose the following values: $c_\star = 32$ and $s_\star = 1\,400$. This globally corresponds to the point of coordinates $(2.5, 1.5)$ in the two subfigures, which presents a good compromise between a reasonably high granularity (with superpixels of approximately 100 pixels) and a low discordance (around 0.06). The induced partitions for the three datasets are exemplified in Figure 4.

6.3. Construction Features

The construction of a BPT requires to choose features involved in the definition of the region model and a merging criterion for assessing the relevance of merging two nodes / regions.

6.3.1. Region Model – Color and Geometry Features

For defining the region model that describes each region / node of the BPT, we first consider colorimetric features. Such features are classically chosen as they provide necessary criteria for merging two nodes (i.e. two nodes with very different color values will not be merged in priority). For a given node N, let us consider two characteristics:

- the range of the pixel intensity values for each image channel $[v_b^-(N), v_b^+(N)]$, where $v_b^\star$ provides the extremal values for the b-th spectral band in I (i.e. in I_b);
- the mean of the pixel intensity values for each image channel $\mu_b(N) = \frac{1}{|N|}\sum_{x=1}^{|N|} I_b(x)$.

We also consider simple shape features as proposed in [46]. A simple yet important one takes into account the region / node area $|N|$ in the region model since size independent color-based measures tend to build trees with few large nodes and a large number of extremely small ones. When involving region area information (noted $\mathcal{A}$) during the creation of the nodes of the tree, the cost of merging for small regions decreases, forcing small regions / nodes to be merged together first and promoting the creation of larger nodes.

Another geometric information that can be considered is the contour of the regions. As we focus on natural images, most "semantic" objects are compact and regular, leading to quite simple contours. Consequently the analysis of shape complexity, such as the perimeter $\mathcal{P}$ of the regions, can provide additional information for building nodes better corresponding to such objects.

6.3.2. Merging Criterion

For each pair of adjacent regions N_i, N_j, characterized with the 4 features described above ($v_b^\star$, μ_b, $\mathcal{A}$, $\mathcal{P}$), we compute several dissimilarity measures assessing their likeness, involving differently these feature values between pairs of regions. The lower the values of these measures, the more similar the two regions, and the higher the priority for their merging during the BPT construction.

Different merging criteria are then proposed, based on various strategies and involving different combinations of features, that lead to the construction of different variants of BPTs:

- The first purely-colour merging criterion function O_{colour} is defined as the increase of the range of the pixel intensity values for each radiometric

band, induced by the putative fusion of incident regions:

$$O_{colour}^{min\text{-}max}(N_i, N_j) = \frac{1}{3}\sum_{b=1}^{3} \max\{v_b^+(N_i), v_b^+(N_j)\} - \min\{v_b^-(N_i), v_b^-(N_j)\} \tag{20}$$

- Regarding mean-based colour merging criteria, two functions were proposed in [46, (2,4)], which assess the deviation between the mean values of each of the two nodes compared to the union of both:

$$O_{colour}^{MSE}(N_i, N_j) = \frac{1}{|N_i| + |N_j|} \sum_{x \in N_i \cup N_j} \|I(x) - \mu(N_i \cup N_j)\|_2^2 \tag{21}$$

$$O_{colour}^{WSDM}(N_i, N_j) = \sum_{k \in \{i,j\}} |N_k| . \|\mu(N_k) - \mu(N_i \cup N_j)\|_2^2 \tag{22}$$

  with $\mu(N) = \frac{1}{|N|}\sum_{y \in N} I(y)$.

- As mentioned above, it is also possible to consider a term related to the contour complexity of the merged regions, as proposed in [46]. The adopted measure computes the increase in perimeter of the new region with respect to the largest of the two merged regions: $\Delta P(N_i, N_j) = \min(\mathcal{P}_i, \mathcal{P}_j) - 2\mathcal{P}_{i,j}$ where $\mathcal{P}_i$, $\mathcal{P}_j$ are the perimeters of N_i, N_j, respectively, and $\mathcal{P}_{i,j}$ is the length of the frontier between the regions. The term that measures contour complexity is then:

$$O_{contour}(N_i, N_j) = \max\{0, \Delta P(N_i, N_j)\}$$

  that sets to 0 negative increments that occur for instance when a region is nested by the other. Color and contour similarity measures are linearly combined as:

$$O_{col\text{-}cont}(N_i, N_j) = \alpha O_{colour}(N_i, N_j) + (1 - \alpha) O_{contour}(N_i, N_j) \tag{23}$$

  In our study, this trade-off value α is empirically set based on preliminary experiments. It is worth mentioning that this parameterization is particularly difficult since a normalization is required in order to make O_{colour} and $O_{contour}$ comparable in terms of values, whereas the actual choice of α already requires to run the method many times.

6.4. Compared Frameworks

In order to emphasize the differences of behaviour between our framework and other (close) frameworks proposed in the literature, we propose some adaptations of the following frameworks, since they are not directly applicable under the given hypotheses of object segmentation.

- The framework proposed in [33] relies on object and background markers, and aims at computing the largest nodes including the object marker whereas not intersecting the background marker. Such framework can be adapted to our hypotheses. In particular, we can build from a binary ground-truth $G \subseteq \Omega$ two markers G_F and G_B for the foreground and background (by eroding them), respectively, defined as $G_F = G \ominus E$ and $G_B = (\Omega \setminus G) \ominus E$ for a given structuring element E. For such set of markers, we can compute the largest nodes that include G_F without intersecting G_B (which is equivalent to computing the cut with the minimal number of nodes for such property). By considering increasing sets of structuring elements E, starting from $E = \emptyset$, it is then possible to study the evolution of the F-measure of the segmented object vs. the associated ground-truth, with respect to the size of E.

- The framework proposed in [30] aims at computing the optimal cut of a tree for a given quality metric that allows to compute such optimal cut from the union of the optimal cuts of the two subtrees. Such a framework is initially proposed for partition comparison metrics, but it remains valid for other kinds of metrics, and in particular metrics for object vs. cut comparison. As initially observed in [30], providing the only optimal cut is not sufficient for assessing the quality of the segmentation, since such cut may optimize the chosen metric whereas being composed of (too) many nodes. In such case, this optimal cut and the associated optimal metric value do not necessarily characterize the quality of the BPT. For implementation purpose, we slightly modified this framework based on our approach proposed in Section 5.3. More precisely, instead of computing the best cut, we compute the best cuts for each cardinality within $[0, p]$, $p > 0$, by assuming that a good cut provides not only a good metric value, but also has a low cardinality, as initially suggested in [30]. We consider this slightly modified framework for the quality metric defined in (19).

6.5. Results

In this section, we present the results of our proposed evaluation framework and, as comparative study, those of [33] and [30] for BPTs built from the images contained in the three datasets: Grabcut, Weizmann and VOC. We considered six kinds of BPTs, built with the previously mentioned criteria, plus a random one which is used as baseline:

- As a naive baseline, we first build a random BPT. To this end, we choose, at each iteration, a pair of nodes to merge randomly, among all the pairs of adjacent nodes. No colorimetric proximity information is thus considered. The underlying idea is to highlight the behavior of the chosen merging criteria (and region model) and the limit scores of the quality criteria proposed in this work;
- Min-max (20);
- MSE (21);
- WSDM (22);
- Min-max + contour (20, 23);
- MSE + contour (21, 23);
- WSDM + contour (22, 23).

The metrics chosen for evaluation purposes of the seven kinds of BPTs are:

- C_1, C_2, Q_3 (5, 6, 9) for the proposed intrinsic analysis (the metrics Q_1 and Q_2 (7, 8) are not considered here due to the large size of the ground-truth objects with respect the size of the images; under such assumption, both metrics do not provide relevant information on the quality of the BPTs);
- the F-measure (i.e. the harmonic mean of precision and recall) of the largest object vs. the ground-truth, and its evolution with respect to the size of the structuring element for [33];
- the true positives minus false positives measure (noted TPFP) (19), normalized by the size of the ground-truth, and its evolution with respect to the size of the optimal cut for [30];
- the Dice measure (17) for the proposed extrinsic analysis.

Table 1: Measures C_1, C_2, Q_3 (mean values) for the intrinsic analysis of the BPTs built from the images of the Grabcut dataset (see Sec. 6.5.1). The best scores are higlighted in bold.

Grabcut

	C_1	C_2	Q_3
Random	0.163	0.262	0.737
Min-max	**0.629**	**0.701**	**0.299**
MSE	0.453	0.528	0.472
WSDM	**0.636**	**0.707**	**0.293**
Min-max + contour	0.519	0.572	0.427
MSE + contour	0.491	0.612	0.388
WSDM + contour	**0.631**	**0.707**	**0.290**

Table 2: Measures C_1, C_2, Q_3 (mean values) for the intrinsic analysis of the BPTs built from the images of the Weizmann dataset (see Sec. 6.5.1). The best scores are higlighted in bold.

Weizmann

	C_1	C_2	Q_3
Random	0.190	0.277	0.720
Min-max	**0.741**	**0.815**	**0.186**
MSE	0.549	0.613	0.387
WSDM	**0.715**	**0.843**	**0.157**
Min-max + contour	0.691	0.770	0.230
MSE + contour	0.599	0.687	0.312
WSDM + contour	**0.709**	**0.834**	**0.166**

Table 3: Measures C_1, C_2, Q_3 (mean values) for the intrinsic analysis of the BPTs built from the images of the VOC dataset (see Sec. 6.5.1). The best scores are higlighted in bold.

VOC

	C_1	C_2	Q_3
Random	0.135	0.232	0.766
Min-max	0.521	0.549	0.447
MSE	0.420	0.392	0.608
WSDM	**0.586**	**0.594**	**0.403**
Min-max + contour	0.422	0.430	0.567
MSE + contour	0.480	0.526	0.474
WSDM + contour	**0.597**	**0.617**	**0.383**

6.5.1. Intrinsic Analysis

For each one of the three datasets, the mean values of the C_1, C_2, Q_3 measures were computed for each of the seven kinds of BPTs built from the images. The results are reported in Tables 1–3.

6.5.2. Evaluation Framework Adapted From [33]

For each one of the three datasets, the F-measure values obtained with the adapted framework from [33] are provided for foreground / background markers generated by erosion with structuring elements of increasing sizes (we consider a square structuring element of size $2k+1$ with increasing values $k \in [0, 10]$). The

closer the results to 1, the better the segmentation. The results are provided in Figure 6 for each of the seven kinds of BPTs built from the images.

6.5.3. Evaluation Framework Adapted From [30]

For each one of the three datasets, the mean TPFP value of the optimal cuts of size s varying in $[0, 30]$ are computed for the seven ways of building BPTs. In order to make these mean values meaningful, the TPFP values are normalized, for each image, by the size of the ground-truth. These optimal TPFP values are computed following the algorithmic process proposed in Section 5.4, following the framework developed in [30]. The results are reported in Figure 7.

6.5.4. Extrinsic Analysis

We finally compute the Dice scores for the three datasets and the seven ways of building BPTs. To this end, we rely on the approximate approach detailed in Section 5.5 in order to estimate these scores without computing them exhaustively on the whole space of possible cuts.

In order to carry out a reasonably low number of computations for each random set of cuts, we perform a seeded random search. More precisely, for each BPT, we determine 1 000 000 cuts subdivided into 10 000 series of 100 cuts. Each series of 100 cuts is built from a seed cut, namely the cut that optimizes the TPFP measure, by iteratively modifying randomly the contribution of exactly one leaf to the segmentation (addition or removal) with respect to the previous cut. In other words, we randomly build 10 000 paths of length 100 within the space of segmentations starting from this seed, in order to explore the space of potential segmentation more densely around a region that is assumed to be close to the optimum (since Dice and TPFP, although different, present close properties). As in Section 6.5.3, we store the best Dice score for each size of cut associated to the segmentations. The mean results for each dataset and each feature are provided in Figure 8.

6.6. Discussion

The results obtained from both our intrinsic / extrinsic frameworks and the two frameworks adapted from [33] and [30] lead globally to the same conclusions (note that the framework [30] is also adapted based on our own extrinsic framework). They emphasize the superiority of WSDM, which is slightly better than Min-Max and significantly better than MSE. They also show that, for a given metric, the BPTs contain more relevant nodes for Grabcut and Weizmann than

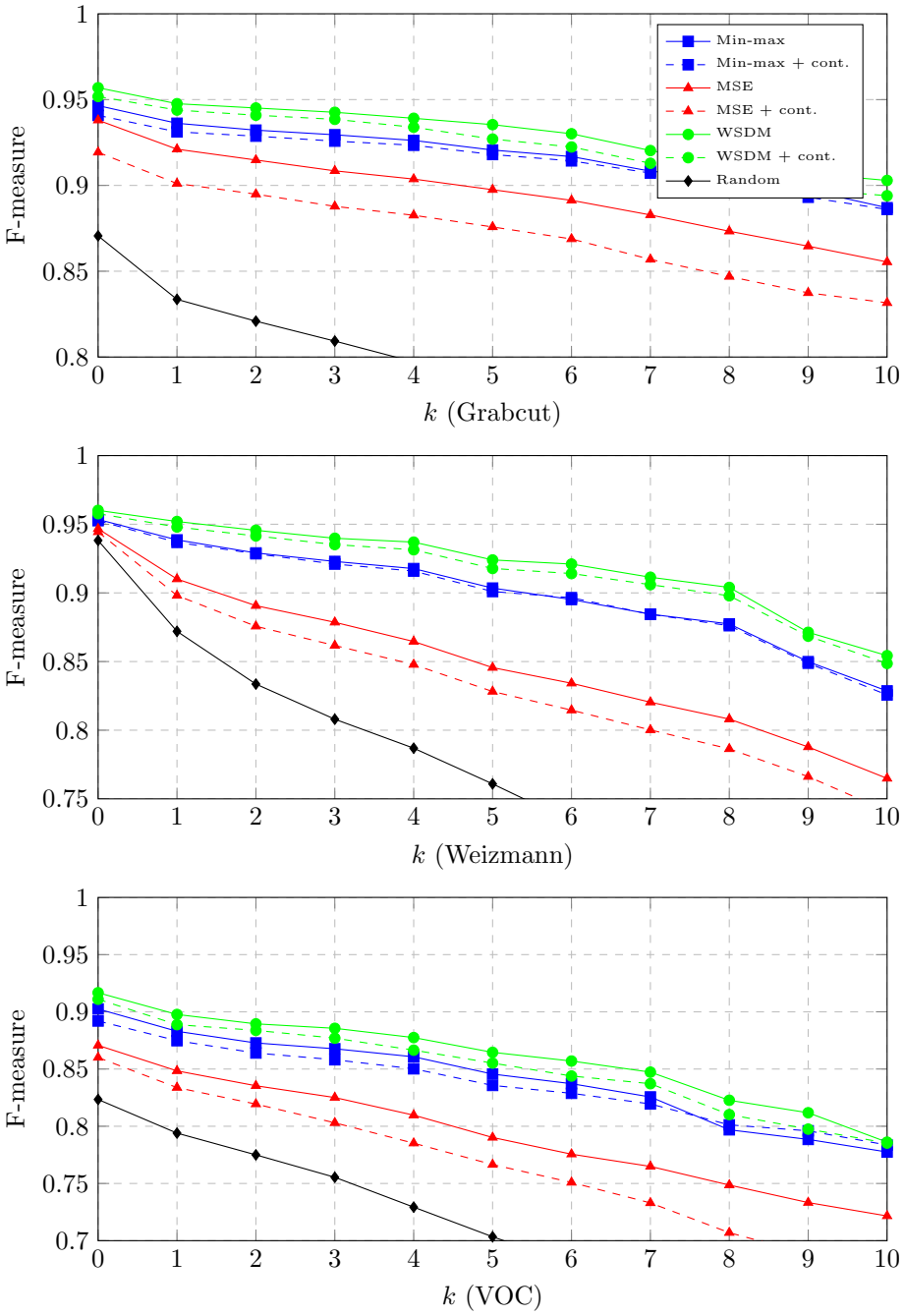

Figure 6: F-measure of the segmentations obtained from the seven kinds of BPTs, with respect to the size $2k+1$ of the structuring elements used for erosion (background and foreground) of the ground-truth (see Sec. 6.5.2). Top: Grabcut. Middle: Weizmann. Bottom: VOC. BPTs built with Min-max (blue, squares), MSE (red, triangles) and WSDM criteria (green, disks) and random (black, diamonds) with (dashed) or without additional contour information (plain) for Min-max, MSE and WSDM.

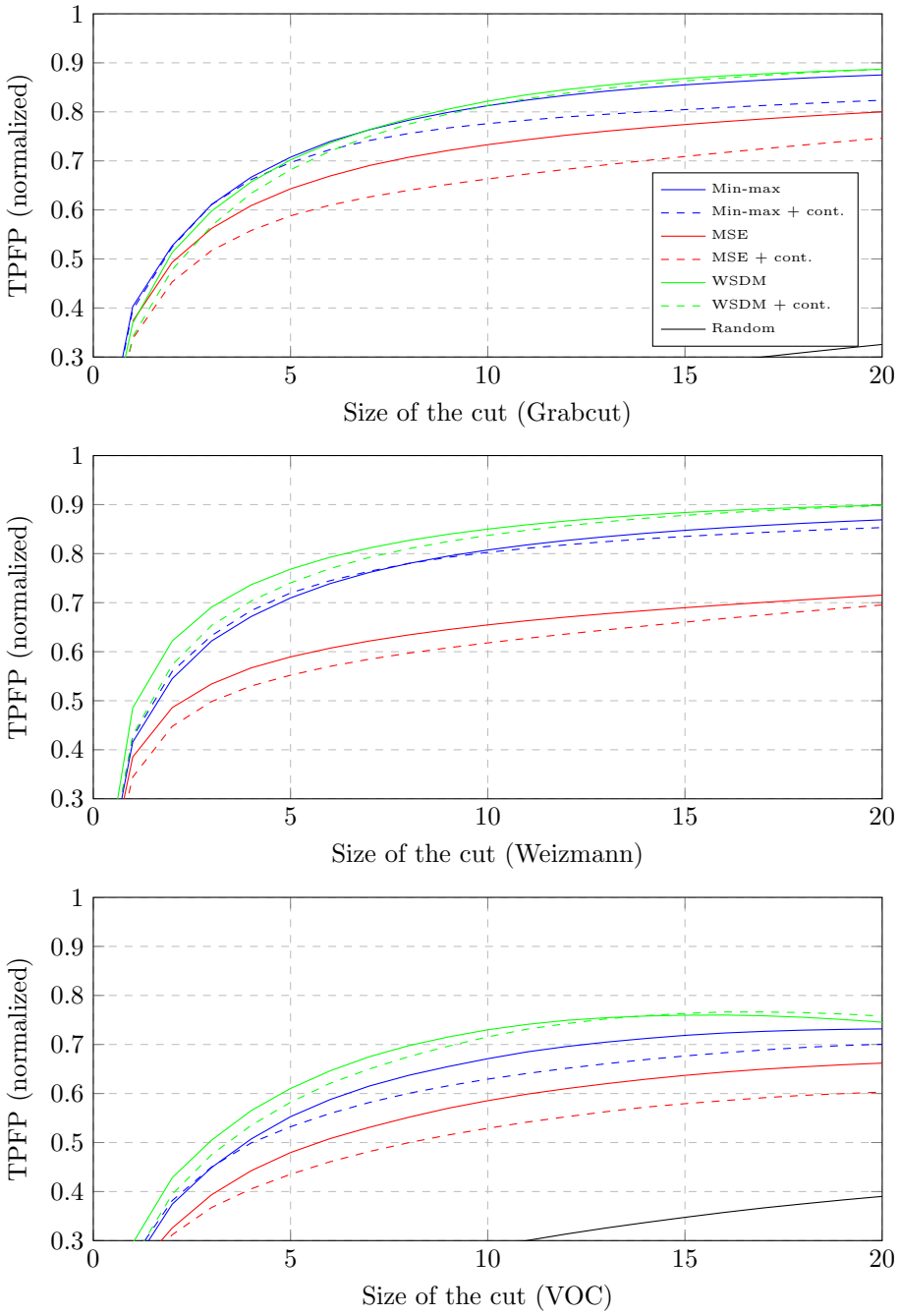

Figure 7: Normalized (mean) value of the TPFP measure for the optimal cuts of a given size from the seven kinds of BPTs, with respect to the size of the cut (see Sec. 6.5.3). Top: Grabcut. Middle: Weizmann. Bottom: VOC. BPTs built with Min-max (blue), MSE (red) and WSDM criteria (green) and random (black) with (dashed) or without additional contour information (plain) for Min-max, MSE and WSDM.

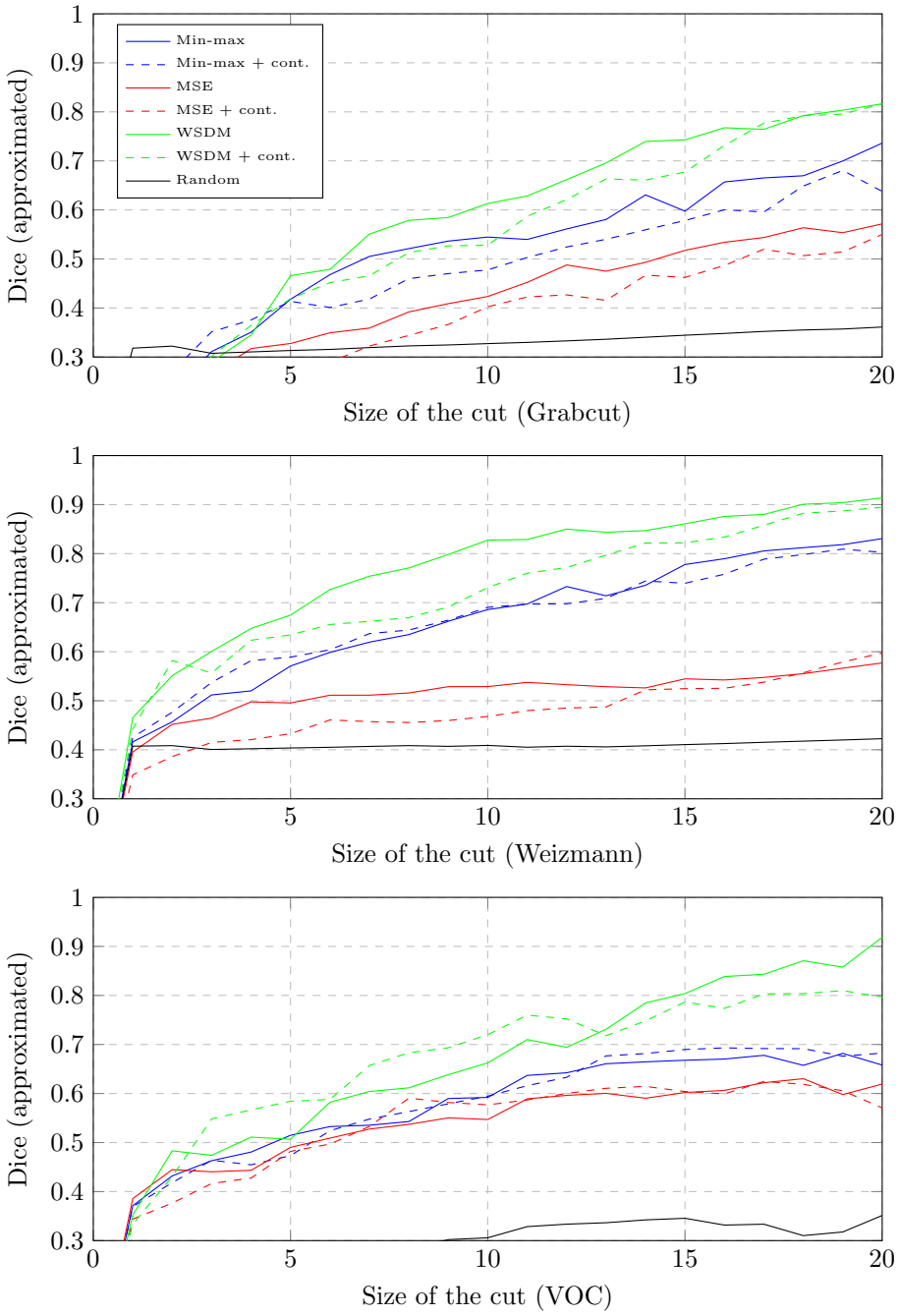

Figure 8: Normalized (mean) value of the Dice measure for the (near-)optimal cuts of a given size from the seven kinds of BPTs, with respect to the size of the cut (see Sec. 6.5.4). Top: Grabcut. Middle: Weizmann. Bottom: VOC. BPTs built with Min-max (blue), MSE (red) and WSDM criteria (green) and random (black) with (dashed) or without additional contour information (plain) for Min-max, MSE and WSDM.

for VOC, which is indeed a dataset with more complex images (Weizmann also appears as a little bit more challenging than Grabcut).

These results are qualitatively confirmed by a visual inspection of horizontal cuts of the BPTs for WSDM, Min-Max and MSE for three images representative of the three datasets, illustrated in Figures 9–11. In these images, we can observe that WSDM better preserves the structure of the car, in Figure 9, compared to Min-Max and a fortiori MSE, proposing a more degraded result. For the tree in Figure 10, WSDM and Min-Max provide satisfactory results, where the tree is not fused with the background of the image, whereas MSE is not able to preserve this separation. The behaviour of Min-Max is, however, less homogeneous than WSDM, with large regions of the tree that hardly fuse with small regions. In Figure 11, that corresponds to the most challenging dataset, WSDM and Min-Max also behave better than MSE, which rapidly fuses the background with parts of the motorcycle. By contrast, WSDM and Min-Max better preserve the motorcycle, with however improved details in WSDM. One has to keep in mind that these examples illustrate horizontal cuts of the BPTs, whereas it is of course possible to define non-horizontal cuts for building segmentations. However, these horizontal cuts characterize the global behaviour of the BPTs construction, and their ability to allow more or less easily to discriminate objects of interest. This visual inspection then confirms the consensual conclusions of the different tested frameworks.

It also appears that for a given metric, the version without additional contour is always better than the version with contours. This tends to confirm the fact that it is indeed difficult to easily normalize two submetrics of distinct semantics and define a trade-off parameter between them in order to build a mixed metric for BPT construction. This difficulty to easily build mixed metrics is a well-known fact in image processing approaches based on the optimization of an energy. This fact had led us to investigate alternative ways of building BPTs from multiple features in [12].

From a general point of view, it appears that the four frameworks, namely our intrinsic and extrinsic ones, and the two frameworks adapted from [33] and [30] constitute relevant approaches for assessing the quality of a BPT, since they provide homogeneous results, that seem to be confirmed by a visual inspection of the data. Our intrinsic framework has the virtue to provide scalar results, that can be easily interpreted, and not to require additional computation time, since these scalar results are obtained from values directly computed during the construction of the BPT. The other three frameworks require the inspec-

tion of chosen metrics (here, F-measure, TPFP and Dice) and their evolution with respect to the size of the cuts computed from the BPT. In particular, the framework proposed by [30] can only deal with specific metrics, for which the computation of the optimal cut is expressed a separable problem. The framework adapted from [33] can deal with any metrics, but requires the computation of background and foreground markers, which can be problematic when the ground-truth are small objects (which was not the case here). Our extrinsic framework deals with separable metrics, and does not require such markers, but it requires the computation of a large set of cuts, but with a low computational cost, in particular sublinear compared to the size of the investigated subtree. One can observe however, that a limited number of cuts can be sufficient if the random search is initialized in relevant parts of the research space. Here, investigating 10^6 cuts per BPT (compared to the exponential number of possible cuts) was indeed sufficient to conclude, despite approximated results that explain the irregular shape of the computed curves (see Figure 8) compared e.g. to the smooth shape of Figure 7.

As a conclusion, it appears that the proposed intrinsic and extrinsic frameworks provide relevant results, that corroborate the evaluations carried out from the frameworks adapted from [33] and [30]. They also present properties which are complementary to these existing / adapted frameworks, providing a novel contribution to the state of the art of BPT analysis in the context of object segmentation.

7. Concluding Remarks

7.1. Direct Extensions

The extrinsic quality analysis process discussed here has been presented for one given ground-truth example G. However it can involve many ground-truth examples for the analysis of a BPT on a given image, without loss of generality. In particular, if various examples $G^i \subset \Omega$ are provided, it is generally sufficient to choose them sufficiently distant from each other for assuming that their global processing is equivalent to the union of their individual processings, as their respective partial cut may then not interact.

It is also worth mentioning that the proposed process allows for uncertainty handling. On the one hand, defining the G^i in a fuzzy way is tractable, since they are only involved in the computation of Λ. On the other hand, even for crisp G^i examples, the use of metrics Λ involving uncertainty (e.g. via distance

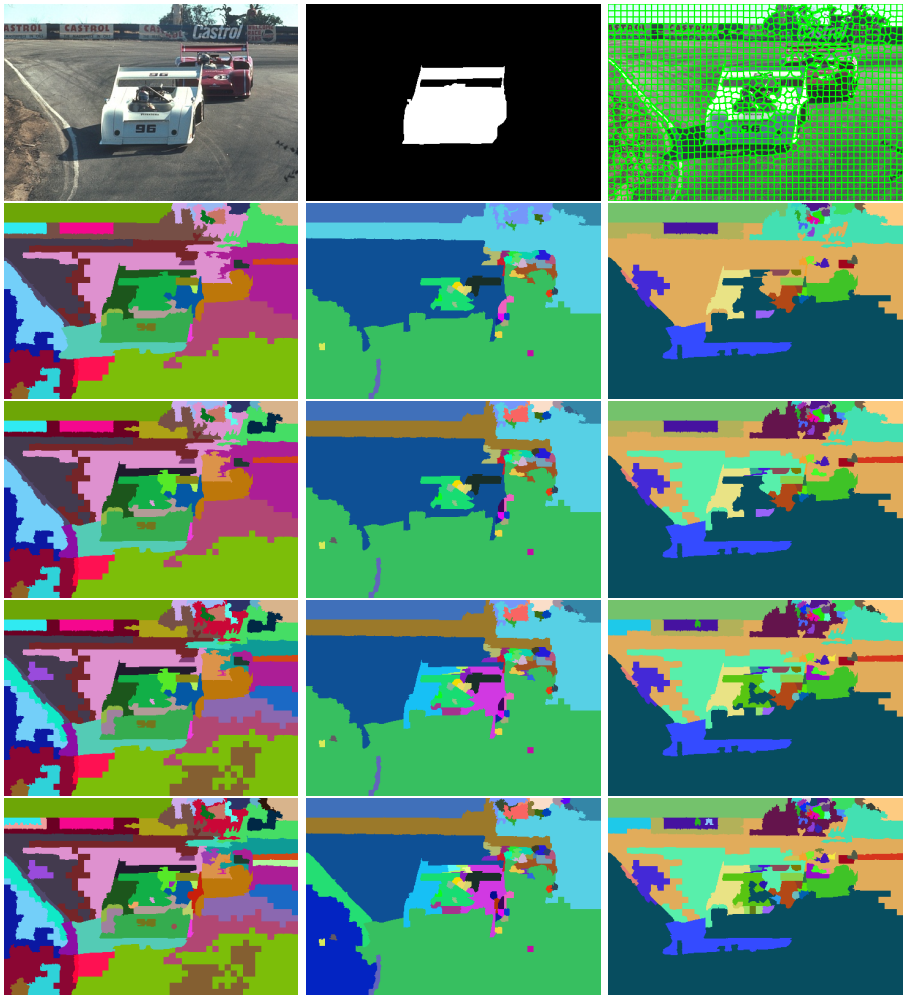

Figure 9: Horizontal cuts of the BPTs obtained from the image of Figure 4 (first row), representative from the Grabcut dataset. First line: image, ground-truth and initial SLIC partition. First column: BPT built with WSDM. Second column: BPT built with Min-Max. Third column: BPT built with MSE. From second to fifth rows: horizontal cuts with 40, 50, 60, and 70 nodes. Each node is represented with a false colour, for the sake of visualization.

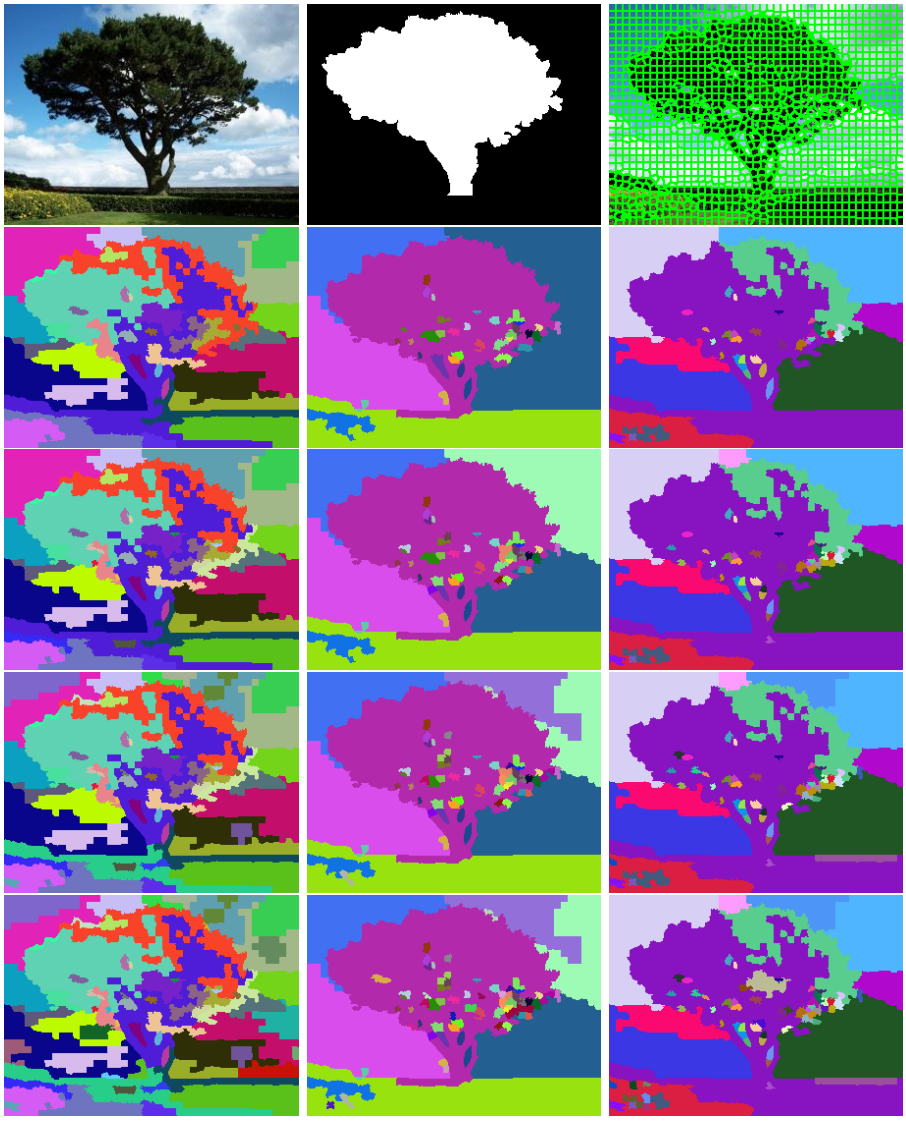

Figure 10: Horizontal cuts of the BPTs obtained from the image of Figure 4 (first row), representative from the Weizmann dataset. First line: image, ground-truth and initial SLIC partition. First column: BPT built with WSDM. Second column: BPT built with Min-Max. Third column: BPT built with MSE. From second to fifth rows: horizontal cuts with 40, 50, 60, and 70 nodes. Each node is represented with a false colour, for the sake of visualization.

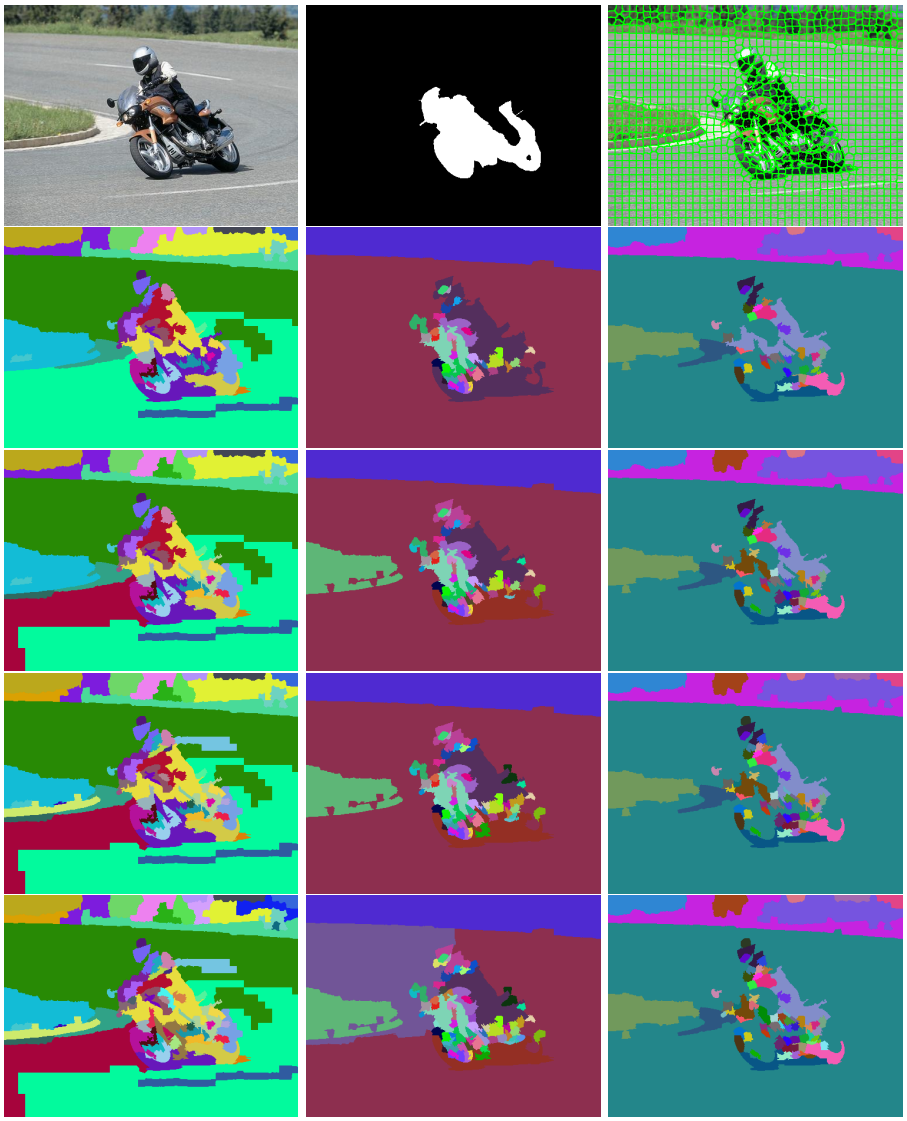

Figure 11: Horizontal cuts of the BPTs obtained from the image of Figure 4 (first row), representative from the VOC dataset. First line: image, ground-truth and initial SLIC partition. First column: BPT built with WSDM. Second column: BPT built with Min-Max. Third column: BPT built with MSE. From second to fifth rows: horizontal cuts with 40, 50, 60, and 70 nodes. Each node is represented with a false colour, for the sake of visualization.

maps from the borders of G^i) remains compliant with the proposed algorithmic process.

Finally, as one quality score $\Lambda_{G^i}(\widehat{S^i})$ can be obtained for each ground-truth example G^i, it is possible to combine these individual results in order to obtain a global quality score $\Lambda_{G^\star}(\widehat{S^\star})$ of the BPT with respect to the whole set of these ground-truth examples.

The very first idea for defining $\Lambda_{G^\star}(\widehat{S^\star})$ is indeed to compute an average value of all the local quality scores, i.e. $\Lambda_{G^\star}(\widehat{S^\star}) = 1/k \sum_{i=1}^{k} \Lambda_{G^i}(\widehat{S^i})$. However, each $\mathfrak{G}^i$ may be endowed with a given semantic label. In such case, it may be important to consider this semantic information to avoid potential bias effects. In particular, it is relevant to model the relative importance of each semantic class C. In complex cases, it may be also relevant to model the relative importance of each segment within each class.

To this end, we can define a weighted formulation of the global quality score as:

$$\Lambda_{G^\star}(\widehat{S^\star}) = \sum_{C} w_C \sum_{G^i \in C} w_i \Lambda_{G^i}(\widehat{S^i})$$

with $\sum_C w_C = 1$ and $\sum_{G_i \in C} w_i = 1$. The weights w_C can be used to assess the relative importance of each semantic class C. The weights w_i can be used for normalizing the local quality metric or to discriminate the importance of each example. They can also be used for quantifying the relevance / confidence of a ground-truth example.

The use of semantic information for designing the global quality metric also argues in favour of the possible design of nonlinear definitions of $\Lambda_{G^\star}(\widehat{S^\star})$. Two specific formulations can be proposed:

$$\Lambda^{\min}_{G^\star}(\widehat{S^\star}) = \min_{C} \sum_{G_i \in C} w_i \Lambda_{G^i}(\widehat{S^i})$$
$$\Lambda^{\max}_{G^\star}(\widehat{S^\star}) = \max_{C} \sum_{G_i \in C} w_i \Lambda_{G^i}(\widehat{S^i})$$

The first allows to determine BPTs that are able to efficiently characterize *all* the classes of ground-truth examples; the second provides a way to determine BPTs that detect (at least) one among a set of given classes.

7.2. Perspectives

The proposed evaluation framework can be used both in a retrospective or a prospective way. In the prospective way, it can help for the meta-parametrization of BPTs, e.g. for applications involving multifeature BPTs, a

variant recently introduced by [12] for building BPTs from many images and/or many construction metrics. In the retrospective way, it can allow the expert user to assess his/her choices in terms of BPT construction metrics. In particular, he/she can verify, a posteriori, (combination) which metric(s), among several existing ones, is the best for a given segmentation application. In this context, this evaluation framework could be generalized to other hierarchical structures to help their comparison, in particular in the recently field of hierarchical model fusion [47].

Acknowledgements

This work was supported by the French *Agence Nationale de la Recherche* (Grants ANR-15-CE23-0009, ANR-17-CE23-0015 and ANR-18-CE23-0025) and by the American Memorial Hospital Foundation.

References

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

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

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

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

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

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

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

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

[9] B. Perret, S. Lefèvre, C. Collet, E. Slezak, Hyperconnections and hierarchical representations for grayscale and multiband image processing, IEEE Transactions on Image Processing 21 (1996) 14–27.

[10] 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.

[11] P. Soille, Constrained connectivity for hierarchical image partitioning and simplification, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 1132–1145.

[12] J. F. Randrianasoa, C. Kurtz, E. Desjardin, N. Passat, Binary Partition Tree construction from multiple features for image segmentation, Pattern Recognition 84 (2018) 237–250.

[13] J. 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, Vol. 6671 of Lecture Notes in Computer Science, 2011, pp. 306–319.

[14] 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.

[15] 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.

[16] 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.

[17] S. Valero, P. Salembier, J. Chanussot, Hyperspectral image representation and processing with binary partition trees, IEEE Transactions on Image Processing 22 (2013) 1430–1443.

[18] 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.

[19] 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.

[20] 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.

[21] P. Salembier, Study of binary partition tree pruning techniques for polarimetric SAR images, in: ISMM, International Symposium on Mathematical Morphology, Proceedings, Vol. 9082 of Lecture Notes in Computer Science, 2015, pp. 51–62.

[22] S. Valero, P. Salembier, J. Chanussot, Object recognition in hyperspectral images using binary partition tree representation, Pattern Recognition Letters 56 (2015) 45–51.

[23] M. Polak, H. Zhang, M. Pi, An evaluation metric for image segmentation of multiple objects, Image and Vision Computing 27 (2009) 1223–1227.

[24] H. Vojodi, A. Moghadam, A supervised evaluation method based on region shape descriptor for image segmentation algorithm, in: AISP, International Symposium on Artificial Intelligence and Signal Processing, Proceedings, 2012, pp. 18–22.

[25] J. Pont-Tuset, F. Marqués, Measures and meta-measures for the supervised evaluation of image segmentation, in: CVPR, Computer Vision and Pattern Recognition, Proceedings, 2013, pp. 2131–2138.

[26] H. Li, J. Cai, T. Nguyen, J. Zheng, A benchmark for semantic image segmentation, in: ICME, International Conference on Multimedia and Expo, Proceedings, 2013, pp. 1–6.

[27] J. Pont-Tuset, P. Arbelaez, J. Barron, F. Marqués, J. Malik, Multiscale combinatorial grouping for image segmentation and object proposal gener-

ation, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (2017) 128–140.

[28] J. Pont-Tuset, F. Marqués, Supervised evaluation of image segmentation and object proposal techniques, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (2016) 1465–1478.

[29] P. Arbelaez, M. Maire, C. C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (2011) 898–916.

[30] J. Pont-Tuset, F. Marqués, Upper-bound assessment of the spatial accuracy of hierarchical region-based image representations, in: ICASSP, International Conference on Acoustics, Speech and Signal Processing, Proceedings, 2012, pp. 865–868.

[31] J. Pont-Tuset, F. Marqués, Supervised assessment of segmentation hierarchies, in: ECCV, European Conference on Computer Vision, Proceedings, Vol. 7575 of Lecture Notes in Computer Science, 2012, pp. 814–827.

[32] B. Perret, J. Cousty, S. J. F. Guimarães, D. Santana Maia, Evaluation of hierarchical watersheds, IEEE Transactions on Image Processing 27 (2018) 1676–1688.

[33] B. Perret, J. Cousty, J. C. Rivera Ura, S. J. F. Guimarães, Evaluation of morphological hierarchies for supervised segmentation, in: ISMM, International Symposium on Mathematical Morphology, Proceedings, Vol. 9082 of Lecture Notes in Computer Science, 2015, pp. 39–50.

[34] J. F. Randrianasoa, C. Kurtz, É. Desjardin, P. Gançarski, N. Passat, Evaluating the quality of binary partition trees based on uncertain semantic ground-truth for image segmentation, in: ICIP, International Conference on Image Processing, Proceedings, 2017, pp. 3874–3878.

[35] J. F. Randrianasoa, C. Kurtz, É. Desjardin, P. Gançarski, N. Passat, Intrinsic quality analysis of binary partition trees, in: ICPRAI, International Conference on Pattern Recognition and Artificial Intelligence, Proceedings, 2018, pp. 114–119.

[36] P. Cettour-Janet, G. Valette, L. Lucas, H. Meunier, G. Loron, N. Bednarek, F. Rousseau, N. Passat, Hierarchical approach for neonate cerebellum segmentation from MRI: An experimental study, in: ISMM, International

Symposium on Mathematical Morphology, Proceedings, Vol. 11564 of Lecture Notes in Computer Science, 2019, pp. 483–495.

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

[38] D. Stutz, A. Hermans, B. Leibe, Superpixels: An evaluation of the state-of-the-art, Computer Vision and Image Understanding 166 (2018) 1–27.

[39] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, L. Van Gool, SEEDS: Superpixels extracted via energy-driven sampling, in: ECCV, European Conference on Computer Vision, Proceedings, Vol. 7578 of Lecture Notes in Computer Science, 2012, pp. 13–26.

[40] P. Neubert, P. Protzel, Superpixel benchmark and comparison, in: Forum Bildverarbeitung, 2012.

[41] B. R. Kiran, J. Serra, Global-local optimizations by hierarchical cuts and climbing energies, Pattern Recognition 47 (2014) 12–24.

[42] A. Blake, C. Rother, M. A. Brown, P. Pérez, P. H. S. Torr, Interactive image segmentation using an adaptive GMMRF model, in: ECCV, European Conference on Computer Vision, Proceedings, Vol. 3021 of Lecture Notes in Computer Science, 2004, pp. 428–441.

[43] S. Alpert, M. Galun, R. Basri, A. Brandt, Image segmentation by probabilistic bottom-up aggregation and cue integration, in: CVPR, Computer Vision and Pattern Recognition, 2007.

[44] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The Pascal Visual Object Classes (VOC) challenge, International Journal of Computer Vision 88 (2010) 303–338.

[45] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, SLIC superpixels compared to state-of-the-art superpixel methods, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (2012) 2274–2282.

[46] V. Vilaplana, F. Marques, P. Salembier, Binary partition trees for object detection, IEEE Transactions on Image Processing 17 (2008) 2201–2216.

[47] D. Santana Maia, J. Cousty, L. Najman, B. Perret, Properties of combinations of hierarchical watersheds, Pattern Recognition Letters 128 (2019) 513–520.

Appendix A. Principal Notations

- Ω: Set of elements composing the support of an image.
- $\mathbb{V}$: Set of values.
- I: Image ($I : \Omega \to \mathbb{V}$).
- G: Ground-truth defined on I ($G \subseteq \Omega$).
- S: Segmentation defined on I ($S \subseteq \Omega$).
- $\frown$: Adjacency relation.
- $\searrow$: Predecessor or parent-child relation.
- $\mathfrak{G}$: Graph.
- $\mathfrak{T}$: Tree.
- $\mathcal{N}$: Set of nodes of a tree.
- N: Node of a tree ($N \in \mathcal{N}$ and $N \subseteq \Omega$).
- $\mathcal{R}$: Singleton set of the root of a tree ($\mathcal{R} \subseteq \mathcal{N}$).
- $\mathcal{B}$: Set of the nodes of a tree which are neither root nor leaves ($\mathcal{B} \subseteq \mathcal{N}$).
- $\mathcal{L}$: Set of the leaves of a tree ($\mathcal{L} \subseteq \mathcal{N}$).
- L: Leaf of a tree ($L \in \mathcal{L}$).
- $\mathcal{C}$: Cut of a tree ($\mathcal{C} \subseteq \mathcal{N}$) .
- γ: Granularity metric (1).
- δ: Discordance metric (2).
- ℓ: number of leaves.
- u: number of unary nodes.
- b: number of binary nodes.