Hierarchizing graph-based image segmentation algorithms relying on region dissimilarity: the case of the Felzenszwalb-Huttenlocher method

Silvio Guimarães, Yukiko Kenmochi, Jean Cousty, Zenilton Patrocinio, Laurent Najman

Hierarchizing graph-based image segmentation algorithms relying on region dissimilarity

the case of the Felzenszwalb-Huttenlocher method

Silvio Guimarães[a,*], Yukiko Kenmochi[b], Jean Cousty[b], Zenilton Patrocinio Jr.[a], Laurent Najman[b]

[a] *PUC Minas - ICEI -DCC - VIPLAB*
[b] *Université Paris-Est, LIGM, ESIEE Paris - CNRS*

Abstract

The goal of this paper is to present an algorithm that builds a hierarchy of image segmentations from a class of dissimilarity criterions, the main example being the criterion proposed by Felzenszwalb and Huttenlocher which provides an observation scale. Specifically, we propose to select, for each observation scale, the largest not-too-coarse segmentation available in the hierarchy of quasi-flat zones. The resulting hierarchy is experimentally proved to be on par with the segmentation algorithm of Felzenszwalb and Huttenlocher, with the added property that it is now much easier to choose (tune) the scale of observation.

Keywords: Scale set theory, Quasi-flat zone hierarchy, Minimum spanning tree, hierarchical image segmentation, graph-based method

1. Introduction and state-of-the-art

Image segmentation is the process of grouping perceptually similar pixels into regions. A hierarchical image segmentation is a set of image segmentations at different detail levels in which the segmentations at coarser detail levels can be produced from simple merges of regions from segmentations at finer detail levels. Therefore, the segmentations at finer levels are nested with respect to those at coarser levels. The level of a segmentation in the hierarchy is also called a scale of observation. As noted by Guigues *et al.* [13], a hierarchy satisfies two important principles of multi-scale image analysis.

- First, the *causality principle* states that a contour presents at a scale k_1 should be present at any scale $k_2 < k_1$.

*Corresponding author
Email addresses: sjamil@pucminas.br (Silvio Guimarães), yukiko.kenmochi@esiee.fr (Yukiko Kenmochi), jean.cousty@esiee.fr (Jean Cousty), zenilton@pucminas.br (Zenilton Patrocinio Jr.), laurent.najman@esiee.fr (Laurent Najman)

- Second, the *location principle* states that contours should be stable, in the sense that they do neither move nor deform from one scale to another.

Some popular image segmentation methods (such as [12, 25] rely on a so-called "scale of observation" to group the pixels into meaningful regions. The main example that will be studied in the sequel of this paper is the one proposed by Felzenszwalb and Huttenlocher [12]. The method, while being very effective in its own right, does not produce a hierarchy, and users face some major issues while tuning the method parameters.

- First, the number of regions may increase when the scale parameter increases. This should not be possible if this parameter was a true scale of observation: indeed, it violates the *causality principle* of multi-scale analysis. Such unexpected behaviour is demonstrated in Fig. 1.

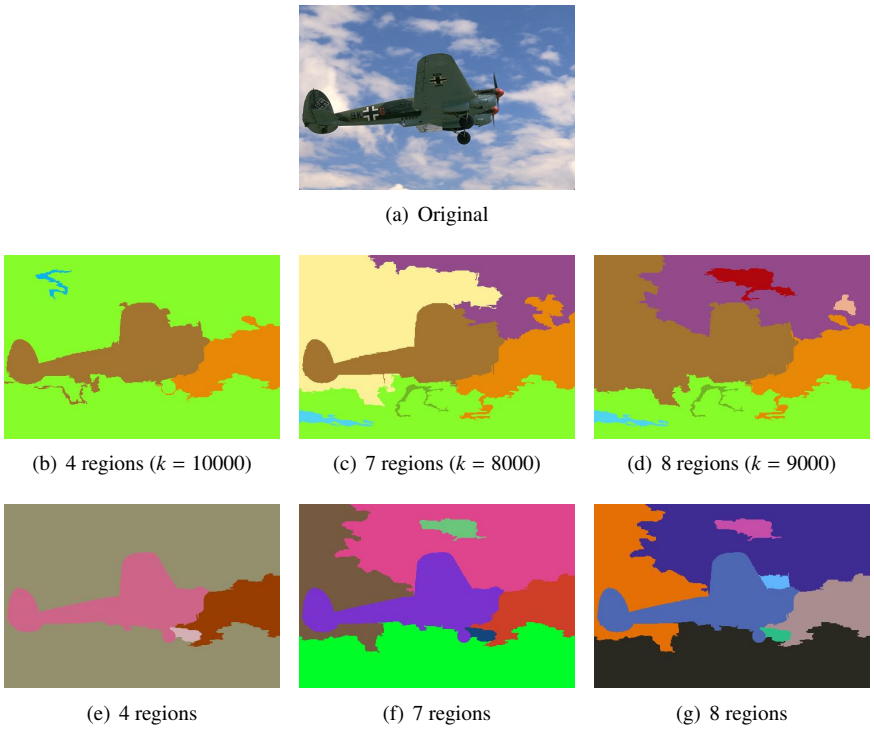
- Second, even when the number of regions decreases, contours are not stable: they can move when the scale parameter varies, violating the *location principle*. Such situations are also illustrated in Fig. 1.

In [10], we study hierarchical segmentation as a generalization of hierarchical classification, the main difference being the connectivity property of a segmentation. In particular, we clarify the links and differences between several different ways of selecting a partition from a hierarchy. Such a formal study provides us with some tools to deal with partitions, segmentations and hierarchies, in a unified framework. Leveraging from this study, we develop in the present paper an efficient methodology[1] for hierarchizing some image segmentation methods that rely on a dissimilarity criterion. Specifically, using the terminology from [12] recalled in the sequel of the paper, we try to select, at each level of the novel hierarchy, the largest not-too-coarse segmentation of the corresponding observation scale amongst all the segmentations available in the quasi-flat zones hierarchy. However, for speed reasons, our algorithm is greedy, and we can not guarantee that the obtained hierarchy is not-too-coarse. Our algorithm has a computational cost similar to the one of [12], but provides the set of segmentations at all observation scales instead of only one segmentation at a given scale. As it is a hierarchy, the result of our algorithm satisfies both the locality principle and the causality principle. Specifically, and in contrast with [12], the number of regions is decreasing when the scale parameter increases, and the contours do not move from one scale to another. This greatly facilitates the selection of a given partition adapted to the application under scrutiny.

The litterature on image segmentation is quite large, and a complete review is beyond the scope of the present paper. Although finding a single partition of an image is still an active topic, it is now recognized that a more robust approach is working in a multi-scale approach that can be given in the form of a hierarchy (amongst many other ones, see for example [3], [31] or [35]).

Any hierarchy can be represented with a tree, specifically with a minimum spanning tree (MST). The first appearance of this tree in pattern recognition dates back to the

[1] This paper is an extension of [15], that proposes a framework to hierarchize GB.



(a) Original

(b) 4 regions (k = 10000) (c) 7 regions (k = 8000) (d) 8 regions (k = 9000)

(e) 4 regions (f) 7 regions (g) 8 regions

Figure 1: Examples illustrating the results of the method proposed by [12] (*p-GB*) and our method (*p-hGB*), obtained from the original image (a): three image segmentations, shown in (b), (c) and (d), are obtained by the method proposed in [12] (*p-GB*) when the observation scale k is set such that k = 10000, 8000, 9000 that lead to 4, 7 and 8 regions, respectively. The area parameter is set to 0.5%, and there is no smoothing. In Figures (b), (c) and (d), the number of regions is not monotonic, when k increases, and the contours between two different k are clearly not stable; they illustrates the violation of the causality and location principles. In contrast, three image segmentations, shown in (e), (f) and (g), are extracted from the hierarchy computed by our method (*p-hGB*) by removing the edges with highest weight values until we obtain the desired number of regions: both causality and location principles are respected.

seminal work of Zahn [38]. Lately, its use for image segmentation was introduced by Morris *et al.* [23] in 1986. In 2004, both Felzenszwalb and Huttenlocher [12] and Nock and Nielsen [25] proposed a statistical image-segmentation method in which the pixel-merging order is similar to the creation of a MST. In [12], the merging sequence relies on both an internal (to the region) constrast measure and an external one. However those region-merging methods [12, 25] do not produce a hierarchy. Several authors [14, 17, 18] proposed to build a hierarchy based on both internal and external contrast measures.

Rather than trying to directly build the optimal hierarchy, a current trend in computer vision is to modify a first hierarchy into a second one, putting forward in the process the most salient regions. A seminal work in that direction is the one of Guigues *et al.* [13], which find optimal cuts in the first hierarchy. This work has been extended in

several directions, see for example [5] and [19]. It is shown in [24] that mathematical morphology provides tools and operators to modify hierarchies, in a spirit similar to what is achieved in [13]. In fact, hierarchies can be seen as a graph on which we can apply any graph-based operator [37], and the well-known watershed operator is itself related to a MST [8]. However, such approaches can not deal with criterion such as the one proposed in [12] for the merging of regions.

In order to go beyond, we explored in [10, 9] the links between mathematical morphology and hierarchical classification. One of the main results of this study is a toolbox allowing us to manipulate a hierarchy. In the present paper, we apply this toolbox to deal with more complex region-merging criteria, the main example being the one of [12].

This paper is organized as follows. In Section 2, based on the theoretical framework proposed in [10], we introduce some formalism for dealing with graphs and hierarchies. We use this formalism in Section 3, to describe our hierarchizing strategy for graph-based segmentations. Many dissimilarity criterions can be used in our proposal, but, in order to clarify and exemplify our algorithm, we use the region dissimilarity proposed in [12]. Using this criterion allows us to perform an experimental study, that is described in Section 4. The main conclusion that can be drawn from this study is that the proposed approach performs similarly to the one of [12], with the significant advantage of being much easier to tune. We finally propose some further research direction in the conclusion (Section 5).

2. Basic notions

In this section, necessary notions for such hierarchical graph-based segmentations are presented, together with properties, on which our algorithm is based. For this purpose, we follow the presentation given in [10].

2.1. Hierarchy of partitions

Given a finite set V, a *partition* of V is defined as a set $\mathbf{P}$ of non-empty disjoint subsets of V whose union is V. Any element of a partition $\mathbf{P}$ is called a *region* of $\mathbf{P}$. If $x \in V$, there is a unique region of $\mathbf{P}$ that contains x, denoted by $\mathbf{P}_x$. Given two partitions $\mathbf{P}$ and $\mathbf{P}'$ of V, we say that $\mathbf{P}'$ is a *refinement* of $\mathbf{P}$, denoted by $\mathbf{P}' \preceq \mathbf{P}$, if any region of $\mathbf{P}'$ is included in a region of $\mathbf{P}$. A *hierarchy* on V is a sequence of indexed partitions of V, $\mathcal{H} = (\mathbf{P}_0, \ldots, \mathbf{P}_\ell)$, such that $\mathbf{P}_{i-1} \preceq \mathbf{P}_i$ for any $i \in \{1, \ldots, \ell\}$. A hierarchy $\mathcal{H}$ is called *complete* if $\mathbf{P}_\ell = \{V\}$ and $\mathbf{P}_0 = \{\{x\} \mid x \in V\}$. Unless otherwise stated, all hierarchies considered in this article are complete.

2.2. Graph and connected hierarchy

A *graph* is a pair $G = (V, E)$ where V is a finite set and E is a subset of $\{\{x, y\} \subseteq V \mid x \neq y\}$. Each element of V is called a *vertex* of G, and each element of E is called an *edge* of G. A *subgraph* of G is a graph (V', E') such that $V' \subseteq V$, and $E' \subseteq E$. If X is a graph, its vertex and edge sets are denoted by $V(X)$ and $E(X)$ respectively.

Let x and y be two vertices of a graph G. A *path* from x to y in G is a sequence $(x_0, x_1, \ldots, x_\ell)$ of vertices of G such that $x_0 = x$, $x_\ell = y$ and $\{x_{i-1}, x_i\}$ is an edge of G

for any i in $\{1, \ldots, \ell\}$. The graph G is *connected* if, for any two vertices x and y of G, there exists a path from x to y. Let A be a subset of $V(G)$. The *graph induced by* A in G is the graph whose vertex set is A and whose edge set contains any edge of G which is made of two elements in A. If the graph induced by A is connected, we also say, for simplicity, that A *is connected*. The subset A is a *connected component* of G if it is connected for G and maximal for this property, *i.e.*, for any subset B of $V(G)$, if B is a connected superset of A, then we have $B = A$. In the following, we denote by $\mathbf{C}(G)$ the set of all connected components of G. It is well known that $\mathbf{C}(G)$ is a partition of $V(G)$. This partition is called the *connected-components partition* induced by G. Thus, the set $[\mathbf{C}(G)]_x$ is the unique connected component of G that contains x.

Given a graph $G = (V, E)$, a *partition* $\mathbf{P}$ *of* V *is connected for* G if every region of $\mathbf{P}$ is connected, and a *hierarchy* $\mathcal{H}$ *on* V *is connected for* G if every partition of $\mathcal{H}$ is connected.

2.3. Edge-weighted graph and quasi-flat zone hierarchy

In this article, we handle connected hierarchies by using edge-weighted graphs. For this purpose, we first see that the level sets of such an edge-weighted graph induce a hierarchy of partitions called a quasi-flat zones hierarchy.

Let G be a graph, and w be a map from the edge set of G into the set $\mathbb{R}^+$ of non-negative real numbers. Then, for any edge u of G, the value $w(u)$ is called the *weight* of u (for w), and the pair (G, w) is called an *edge-weighted graph*.

We assume that G is connected. Without loss of generality, we also assume that the range of w is the set $\mathbb{E}$ of all integers from 0 to $|E| - 1$ (otherwise, one could always consider an increasing mapping from the set $\{w(u) \mid u \in E\} \subset \mathbb{R}^+$ into $\mathbb{E}$). We also denote by $\mathbb{E}^\bullet$ the set $\mathbb{E} \cup \{|E|\}$.

Let X be a subgraph of G and λ be a non-negative integer in $\mathbb{E}^\bullet$. The *λ-level edge set* of X (for w), denoted by $w_\lambda(X)$, is defined by

$$w_\lambda(X) = \{u \in E(X) \mid w(u) < \lambda\}, \tag{1}$$

and the *λ-level graph* of X (for w) is defined as the subgraph $w^V_\lambda(X)$ of X such that

$$w^V_\lambda(X) = (V(X), w_\lambda(X)). \tag{2}$$

Then, the connected-components partition $\mathbf{C}(w^V_\lambda(X))$ induced by the λ-graph of X is called the *λ-level partition* of X (for w).

For $\lambda_1, \lambda_2 \in \mathbb{E}^\bullet$ such that $\lambda_1 \leq \lambda_2$, any edge of $w^V_{\lambda_1}(X)$ is also an edge of $w^V_{\lambda_2}(X)$. Thus, any connected component of $w^V_{\lambda_1}(X)$ is included in a connected component of $w^V_{\lambda_2}(X)$. In other words, $\mathbf{C}(w^V_{\lambda_1}(X)) \preceq \mathbf{C}(w^V_{\lambda_2}(X))$ for $\lambda_1 \leq \lambda_2$. Hence, the sequence of all λ-level partitions of X ordered by increasing value of λ such that

$$\mathcal{QFZ}(X, w) = (\mathbf{C}(w^V_\lambda(X)) \mid \lambda \in \mathbb{E}^\bullet)$$

is a hierarchy, called the *quasi-flat zones hierarchy* of X for w. Observe that this hierarchy is complete if X is connected. It is thus seen that a quasi-flat zones hierarchy $\mathcal{QFZ}(X, w)$ is induced by an edge-weighted graph (X, w). Conversely, it is also shown

in [10] that any connected hierarchy $\mathcal{H}$ can be represented by an edge-weighted graph whose associated quasi-flat zones hierarchy is precisely $\mathcal{H}$, by using the notion of a saliency map. As shown in [10], the saliency map of a hierarchy $\mathcal{H}$ is precisely the minimal map whose quasi-flat zones hierarchy is exactly $\mathcal{H}$.

Let us now consider a minimum spanning tree of G with respect to w, denoted by T, which is a subgraph of G, connecting all the vertices of G, with weight less than or equal to the weight of every other subgraph in the same manner. More formally, the subgraph T is a *minimum spanning tree (MST) of* G if:

1. T is connected; and
2. $V(T) = V$; and
3. the weight of T is less than or equal to the weight of any graph X satisfying (1) and (2) (*i.e.*, X is a connected subgraph of G whose vertex set is V),

where the weight of X for w, denoted by $w(X)$, is defined as $w(X) = \sum_{u \in E(X)} w(u)$. It is then shown in [10] that the quasi-flat zones hierarchy $\mathcal{QFZ}(T, w)$ is the same as $\mathcal{QFZ}(G, w)$. It is also proved in [10] (Theorem 4) that T is minimal for this property as well, *i.e.*, for any subgraph X of T, if the quasi-flat zones hierarchy of X for w is the one induced by G for w, then we have $X = T$. Conversely, any minimal graph for this property is a MST of G.

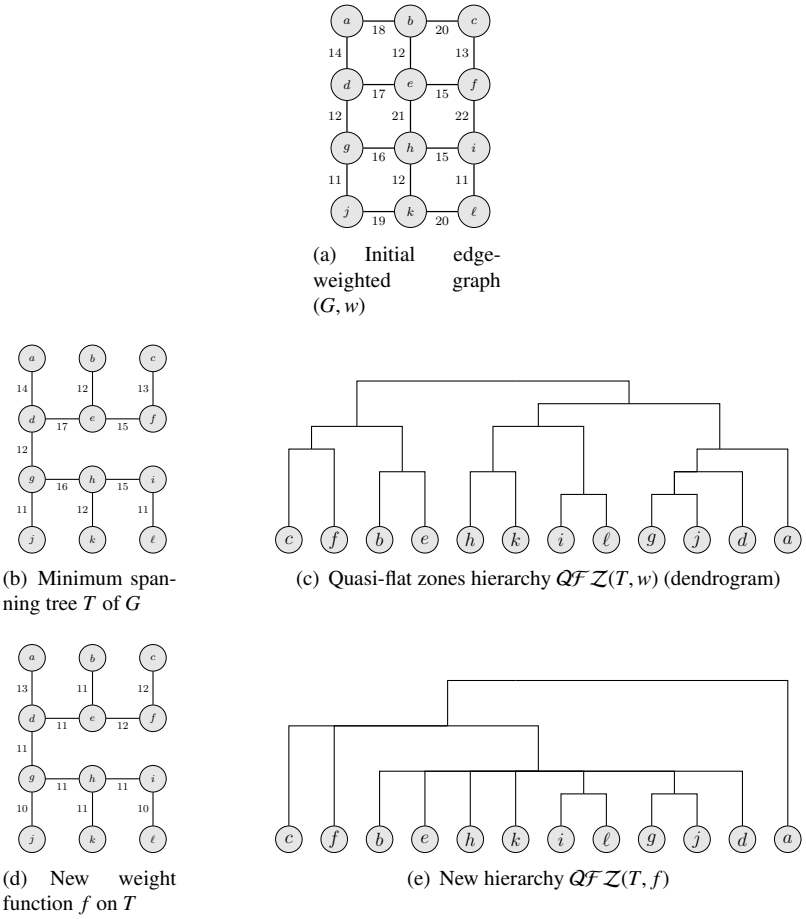
Those results indicate that any connected hierarchy $\mathcal{H}$ for G can be handled by means of a weighted spanning tree which is a subgraph of G. This is indeed the theoretical basis of our algorithm for hierarchical segmentation.

3. Graph-based hierarchical segmentation

3.1. Transformation of hierarchies

Let us suppose that a connected graph $G = (V, E)$ and an associated weight function w are given. As w is given with G, we can also say that a quasi-flat zones hierarchy $\mathcal{QFZ}(G, w)$ is given. The main idea of our algorithm for hierarchical segmentation of V then consists of transforming this initial hierarchy $\mathcal{QFZ}(G, w)$ into another hierarchy by rebuilding the hierarchical structure according to a dissimilarity measure between regions. In fact, as mentioned in the previous section, the hierarchy $\mathcal{QFZ}(G, w)$ is the same as $\mathcal{QFZ}(T, w)$ where T is a MST of G. Therefore, in order to transform the hierarchy $\mathcal{QFZ}(G, w)$ (or $\mathcal{QFZ}(T, w)$), we generate a new weight function f, which can be restricted on T, so that it leads to a new hierarchy $\mathcal{QFZ}(T, f)$. Thus, the main part of our segmentation algorithm is the generation of a new weight function f on T. See Fig. 2 for an example, which illustrates our graph-based hierarchical segmentation. Given the initial edge-weighted graph (G, w) illustrated in Fig. 2 (a), Fig. 2 (c) provides the dendrogram representation of the initial quasi-flat zones hierarchy $\mathcal{QFZ}(T, w)$ generated from the MST T of G for w, illustrated in Fig. 2 (b). After changing the weights on the edges of T, denoted by f, as depicted in Fig. 2 (d), the quasi-flat zones hierarchy is also changed to $\mathcal{QFZ}(T, f)$ whose dendrogram is given in Fig. 2 (e).

While a new weight function f is generated based on a dissimilarity measure between regions, the measure itself is independent from the proposed algorithm. We therefore simply denote it by D in the algorithm, and give its concrete definition for the



(a) Initial edge-weighted graph (G, w)

(b) Minimum spanning tree T of G

(c) Quasi-flat zones hierarchy $\mathcal{QFZ}(T, w)$ (dendrogram)

(d) New weight function f on T

(e) New hierarchy $\mathcal{QFZ}(T, f)$

Figure 2: Example of our graph-based hierarchical segmentation, which consists of transforming an initial hierarchy into another, by changing weights on the edges.

case of the method proposed by [12] later in Section 3.5. The map D associates a value in $\mathbb{E}^\bullet$ to any region pair of a partition $\mathbf{P}$.

3.2. Too-fine/too-coarse partitions

Given a graph $G = (V, E)$, let us consider a partition $\mathbf{P}$ of V and a region pair $A, B \in \mathbf{P}$. If there exists an edge $\{x, y\} \in E$ such that $x \in A$ and $y \in B$, then A and B are said to be adjacent. By taking into account some dissimilarity measure D between adjacent regions, the following notions are considered for partitions. A partition $\mathbf{P}$ is said to be *too fine at level* λ if there exists an adjacent-region pair A, B of $\mathbf{P}$ such that $D(A, B) \leq \lambda$, and *too coarse at level* λ if there exists a proper refinement of $\mathbf{P}$, that is not too fine at level λ. Intuitively, a partition is too fine at level λ if there exist two adjacent regions A and B that should be merged, instead of being separated, since

Algorithm 1: Re-weight

Data: A minimum spanning tree $T = (V, E)$ of an edge-weighted graph (G, w), a dissimilarity measure D

Result: A map f from E to $\mathbb{R}^+$

for *each* $u \in E$ **do** $f(u) := +\infty$;
for *each* $u = \{x, y\} \in E$ *in non-decreasing order for* w **do**
  $f(u) := \min\{\lambda \in \mathbb{R}^+ \mid D([\mathbf{C}(f_\lambda^V(T))]_x, [\mathbf{C}(f_\lambda^V(T))]_y) \leq \lambda - 1\}$

$D(A, B) \leq \lambda$. In contrast, a partition is too coarse at level λ if the splitting of one of its regions leads to a partition which is not too fine, namely, the region should be split. In [12], Felzenswalb *et al.* were interested in partitions of V that are neither too fine nor too coarse for a given λ.

3.3. Too-fine/too-coarse hierarchies

Those notions can be extended to hierarchies of partitions as follows. Let $\mathcal{H} = (\mathbf{P}_\lambda \mid \lambda \in \mathbb{E})$ be a complete hierarchy that is connected for G. We say that $\mathcal{H}$ *is not too fine (resp. not too coarse)* if for any $\lambda \in \mathbb{E}$, the partition $\mathbf{P}_\lambda$ is not too fine (resp. not too coarse) at level λ. By using them, it is natural to look for a possibility of producing hierarchies that are neither too fine nor too coarse.

However, the criterion used in [12] does not straightforwardly lead to a dissimilarity measure for which a hierarchy that is neither too coarse nor too fine always exists. For instance, we can see such a difficulty in the results of the method proposed by [12], shown in Fig. 1 (b-d), which illustrate the violation of the causality and location principles with respect to the observation scales k used in the criterion (more examples can also be found in [15]).

This difficulty motivated us to focus first on hierarchies that are not too coarse (not needed to be not too fine) and such hierarchies always exist, whatever the chosen dissimilarity measure.

The trivial hierarchy, such as the hierarchy whose levels are all the partition of V into singletons, is not too coarse. However, in general, there exist many hierarchies that are not too coarse and one needs to choose among them. One of interesting choices is made by keeping a largest hierarchy among all hierarchies that are not too coarse (see Section 8.3.1 of [10] for further details). In general, finding such a hierarchy is a complex task.

3.4. Algorithm description

The algorithm presented in this article is heuristic and does not guarantee to produce such a not-too-coarse hierarchy whatever the considered dissimilarity measure D. However, reaching this goal for the dissimilarity measure of the segmentation method proposed in [12] guides us to propose Algorithm 1.

Let us suppose that a minimum spanning $T = (V, E)$ of a connected graph G for an associated weight function w is given. The proposed algorithm is based on a merging-region strategy, in which the new weight f on T is calculated iteratively as a sequence

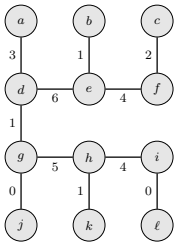


Figure 3: Saliency map $\Phi(\mathcal{H})$ of the hierarchy $\mathcal{H}$ in Fig. 2 (c)

of maps f_i for $i = 0, \ldots, |E|$ so that $f = f_{|E|}$. It is initialized such that

$$f_0(u) = +\infty$$

for every $u \in E$, so that $\mathcal{QFZ}(T, f_0)$ is obviously not too coarse. It is then updated for each edge $u \in E$ one by one in non-decreasing order with respect to the original weight w. Letting L to be the list of such ordered edges, we write $L(i) = u$ for $i = 1, \ldots, |E|$ if the i-th ordered edge is u. Let us suppose that the i-th map f_i is already calculated. Then the $(i + 1)$-th map f_{i+1} is obtained from f_i such that

$$f_{i+1}(u) = \begin{cases} \min\{\lambda \in \mathbb{R}^+ \mid D([\mathbf{C}(f_{i\lambda}^V(T))]_x, [\mathbf{C}(f_{i\lambda}^V(T))]_y) \leq \lambda - 1\} & \text{if } u = L(i+1), \\ f_i(u) & \text{otherwise,} \end{cases}$$

for all $u = \{x, y\} \in E$.

Let $\mathcal{H}_i = \mathcal{QFZ}(T, f_i)$. Then, at any $\lambda \in \mathbb{R}^+$, the partition of $\mathcal{H}_i$ is always coarser than the one of $\mathcal{H}_{i+1}$, namely

$$\mathbf{C}(f_{i\lambda}^V(T)) \preceq \mathbf{C}({f_{i+1}}_\lambda^V(T))$$

for $i = 0, 1, \ldots, |E| - 1$, since $f_i(u) \geq f_{i+1}(u)$ for any $u \in E$. Thus, we can see at the i-th iteration step that the two regions containing respectively x and y of the i-th edge $u = \{x, y\}$ are merged up to the level λ.

We also remark that the algorithm is valid, even though w is not given, if a hierarchy $\mathcal{H}$ is given instead. In fact, it is shown in Section 7 of [10] as well as [11] that any connected hierarchy $\mathcal{H}$ can be represented by an edge-weighted graph whose associated quasi-flat zones hierarchy is precisely $\mathcal{H}$, by using the notion of saliency map $\Phi(\mathcal{H})$. In other words, we can associate to any $\mathcal{H}$ the saliency map $\Phi(\mathcal{H})$ whose quasi-flat zones hierarchy is $\mathcal{H}$. See Fig. 3 for the saliency map $\Phi(\mathcal{H})$ of the quasi-flat zones hierarchy $\mathcal{H}$ illustrated in Fig. 2(c).

3.5. Observation scale dissimilarity

Algorithm 1 requires some dissimilarity measure D. In this article, the observation-scale dissimilarity D, based on the region merging predicate presented in [12], is proposed. Note that this dissimilarity is not defined explicitly in the original method, but it is used implicitly in their predicate for merging regions. In this section, we show how

to extract from their predicate the dissimilarity function, with which we can realize a hierarchical segmentation by using Algorithm 1.

Let us first recall the region-merging predicate used in [12]. It is based on measuring the dissimilarity between two components (*i.e.*, regions) by comparing two types of differences: the inter-component difference and the within-component difference. The first one is defined between two regions C_1 and C_2 by

$$Dif(C_1, C_2) = \max\{w(\{x, y\}) | x \in V(C_1), y \in V(C_2), (x, y) \in E\}$$

and the later one is defined for each component C by

$$Int(C) = \max\{w(\{x, y\}) | x, y \in V(C), \{x, y\} \in E\}.$$

The predicate is then defined by

$$P(C_1, C_2) = \begin{cases} \text{true} & \text{if } \ Dif(C_1, C_2) \leq \min\limits_{i=1,2} \left\{ Int(C_i) + \dfrac{k}{|C_i|} \right\}, \\ \text{false} & \text{otherwise,} \end{cases} \tag{3}$$

where $|C|$ is the set cardinality of C, and k is a given constant parameter.

Le us reformulate (3) in order to obtain a dissimilarity measure D. The merging predicate (3) depends on the value k at which the regions C_1 and C_2 are observed. More precisely, the *observation scale of* C_1 *relative to* C_2 is first defined by

$$S_{C_2}(C_1) = (Dif(C_1, C_2) - Int(C_1))|C_1|,$$

and similarly the one of C_2 relative to C_1 is defined by

$$S_{C_1}(C_2) = (Dif(C_1, C_2) - Int(C_2))|C_2|.$$

Let us define the *observation-scale dissimilarity* between C_1 and C_2 such that

$$D(C_1, C_2) = \max\{S_{C_2}(C_1), S_{C_1}(C_2)\}. \tag{4}$$

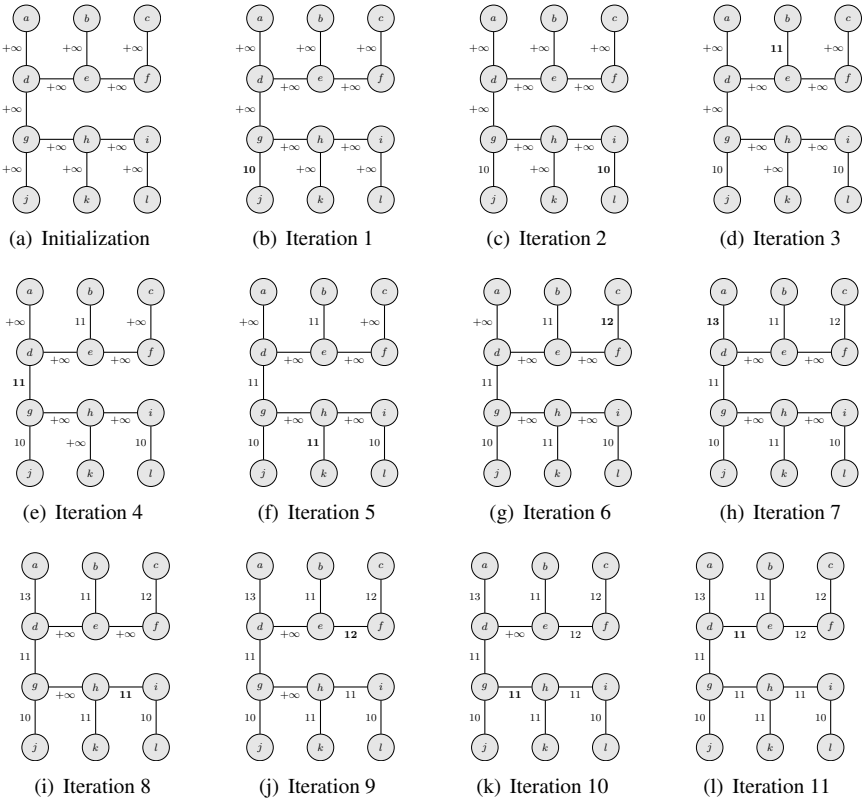
The predicate (3) can then be written using this dissimilarity:

$$P(C_1, C_2) = \begin{cases} \text{true} & \text{if } \ D(C_1, C_2) \leq k, \\ \text{false} & \text{otherwise.} \end{cases}$$

The inequality condition is seen in Algorithm 1, indeed, by replacing $\lambda - 1$ by k.

3.6. Illustration of Algorithm 1

An illustration of Algorithm 1 applied to the minimum spanning tree T of the edge-weighted graph graph (G, w) in Fig. 2 can be found in Fig. 4. After initialization of edge re-weighting with $+\infty$ for all edges in Fig. 4 (a), the new edge weight f for each edge is calculated in the non-decreasing order with respect to the original weight w as shown in in Fig. 4 (b-l). In each step of the iteration, one new edge weight is calculated, depicted as the value in bold. On this example, the interested reader can verify that the hierarchy corresponding to the obtained map shown in Fig. 4 (l) (see also Fig. 2 (e) for its dendrogram representation) is not too coarse.



(a) Initialization (b) Iteration 1 (c) Iteration 2 (d) Iteration 3

(e) Iteration 4 (f) Iteration 5 (g) Iteration 6 (h) Iteration 7

(i) Iteration 8 (j) Iteration 9 (k) Iteration 10 (l) Iteration 11

Figure 4: Illustration of Algorithm 1 with the example in Fig. 2

4. Experiments

The main aim of this section is to compare our method abbreviated by *hGB*, with its original method [12] abbreviated by *GB*. The abbreviations *hGB* and *GB* stand for "the Hierarchical Graph-Based method" and "the Graph-Based method", respectively. To this end, we need to understand the behaviour of *hGB* compared to that of *GB*. Even though both methods are based on the similar observation scale, *hGB* provides a hierarchical structure while *GB* does not. Thus, what we would like to observe in this section is a role of such a hierarchical structure of segmentations in the procedure; indeed, we will see in this section that the hierarchical structure imposed in *hGB* provides as good results as those of *GB*. In addition, *hGB* gives the additional advantage that the hierarchical structure makes easier to choose (tune) the scale of observation corresponding to a desired partition.

For such experiments, both methods are applied to several databases, and comparison analyses between them are made first qualitatively, and then quantitatively. The details including the strategies for the quantitative analysis will be explained in Sections 4.1 and 4.2.

We should mention that the efficient implementation of our method was made by using some data structures similar to the ones proposed in [15]; in particular, the management of the collection of partitions is made using Tarjan's union-find algorithm [33]. Furthermore, we made some algorithmic optimizations to speed up the computations of the hierarchical scales. We also employ a KD-tree for identifying K-nearest neighbours (when it is necessary) in order to create a graph from a given image.

4.1. Compared methods, pre- and post-processing, and underlying graphs

As mentioned above, we compare our method *hGB* that provides a hierarchical graph-based segmentation result with its original method *GB* [12] that provides a non-hierarchical graph-based segmentation result. Note that results of *GB* depend on observation scales, which can be set variously.

The original method *GB* includes the following pre- and post-processing steps:

- Gaussian smoothing of the image with parameter σ as a pre-processing step,
- area-filtering of the segmentation results with parameter τ, which is the ratio of the component size to the image size, as a post-processing step.

In order to make fair comparison, we also apply to our method *hGB* the same pre-processing and a similar, but hierarchical, post-processing, which is using again the technique introduced in [10] and based on re-weighting a spanning tree (see Appendix Appendix A for details). In this article, the two parameter values vary such that $\sigma \in [0, 0.5]$ and $\tau \in [0.001, 0.009]$.

Before applying *GB* and *hGB* methods, it is necessary to transform a given image into an edge-weighted graph. In this article, we consider the following two techniques to get such underlying graphs, similarly to those in [12]:

- **Pixel adjacency graph.** It is induced by the 8-adjacency relation, where each vertex is a pixel and each edge is a pair of adjacent pixels. Two distinct pixels are said to be 8-adjacent if they have a common corner when pixels are represented by squares. Each edge is weighted by a simple color gradient: the Euclidean distance in the RGB space between the colors of the two adjacent pixels. In order to identify this graph type in the following, we will add the letter p followed by the short name of methods (p stands for "pixel" relationship);
- **Nearest-neighbour color-pixel graph**. In this setting, we further add new edges to the pixel adjacency graph. The additional edges are given by the K nearest neighbours of each vertex in a RGBXY space. In this RGBXY space, each pixel is mapped to a 5-tuple (r, g, b, x, y) where r, g and b are the red, green and blue components of the pixel and x and y are the x- and y-coordinates. Then, each edge is weighted by the Euclidean distance in the RGBXY space between the neighbouring pixels. In order to identify this graph type, we will add the letter cp followed by the short name of the method (c stands for "color"). We set $K = 10$ in this article.

4.2. Databases

In order to provide a comparative analysis between *GB* and *hGB*, we used the following three different databases proposed in [1, 2, 3, 29].

BSDS500 The Berkeley Segmentation Dataset [21], called **BSDS500**, is divided in three folds for training, validation and testing, which contain 200, 100 and 200 images, respectively. The training and validation folds are used to set the pre- and post-processing parameters, σ and τ, such that those values lead to the best measure value (see Section 4.5.2 for the detail) on those folds. Each image has a set of 5 to 8 human-marked ground-truth segmentations, which has a high degree of consistency between different human subjects with a certain variance in the level of details. This database is a standard for evaluating hierarchical segmentations [3, 27].

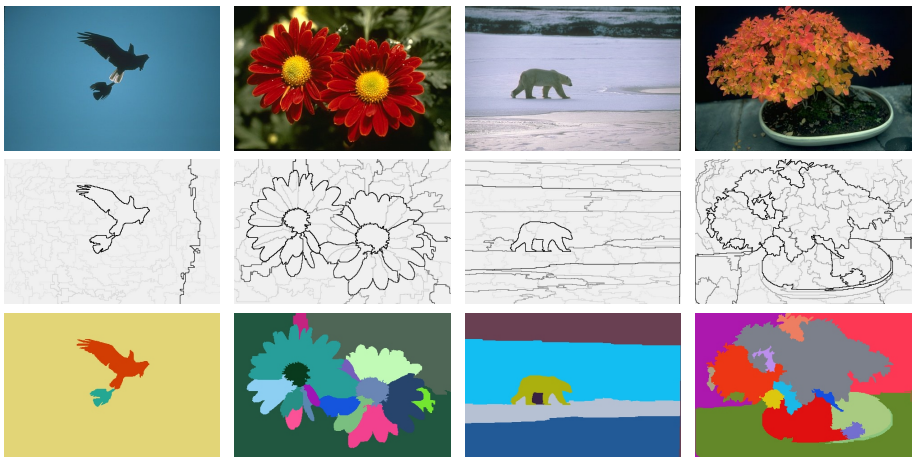
GRABCUT The database proposed in [29], called **GRABCUT**, is originally used for detecting the contour of one object. In fact, some images in this database are also found in **BSDS500**. The images in this dataset are of high quality and contain big objects. All 50 images are used for the parameter setting, and every image has only one ground-truth contour of the one object.

WI1OBJ, WI2OBJ The database proposed in [1, 2] consists of images, each of which contains obviously one or two objects; each object differs from their surroundings by either intensity, texture, or other low level cues. The database is therefore divided into two groups, single and two objects, each of which contains 100 images, called **WI1OBJ** and **WI2OBJ** respectively. As there is no fold in this database, concerning the pre- and post-processing parameter setting, we used all the images. Each image has 1 to 3 human-marked ground-truth regions of the objects of interest. Unlike **GRABCUT**, the images have lower quality and the objects are very small, which can make segmentation results more complicated.

4.3. Qualitative analysis

We first show Fig. 5 to illustrate some results obtained by our method when applied to some images in **BSDS500** using a pixel-based graph. In order to visualize the hierarchical structures $\mathcal{H}$ of our segmentation results, we use saliency maps $\Phi(\mathcal{H})$ [11] as shown in the middle row of the figure: pixel edges (*i.e.* graph edges used in our method) that have higher observation scale dissimilarities are depicted in stronger black. From such a saliency map, the segmentation at a given observation scale can be easily obtained by simple thresholding. The segmentation results of given scales are illustrated in the last row of the figure.

Concerning the comparison between *p-hGB* and *p-GB*, Fig. 1 illustrates some results of *p-hGB* and *p-GB* obtained from the original image (a); (b-d) present the results from *p-GB* containing 4, 7 and 8 regions, while (e-g) present the results from *p-hGB* containing the same numbers of regions. While obtaining the partition from an expected number of regions is easy with our method *p-hGB*, it is quite difficult when using *p-GB*. It should be also noticed that the causality and location principles are missing for the results of *p-GB* in Fig. 1 (b-d) while they are preserved for those of *p-hGB* in Fig. 1 (e-g) thanks to the hierarchical structures.



(a) (b) (c) (d)

Figure 5: Top row: some images from the Berkeley database [3]. Middle row: saliency maps of these images according to our hierarchical method. The number of different segmentations obtained from observation scales of these hierarchies are (a) 240, (b) 443, (c) 405 and (d) 429. Bottom row: examples of segmentations extracted from the hierarchies. The numbers of segmented regions are (a) 3, (b) 18, (c) 6 and (d) 16.

Note that we set $\sigma = 0$ and $\tau = 0.005$ for the pre- and post-processing in both of the experiments.

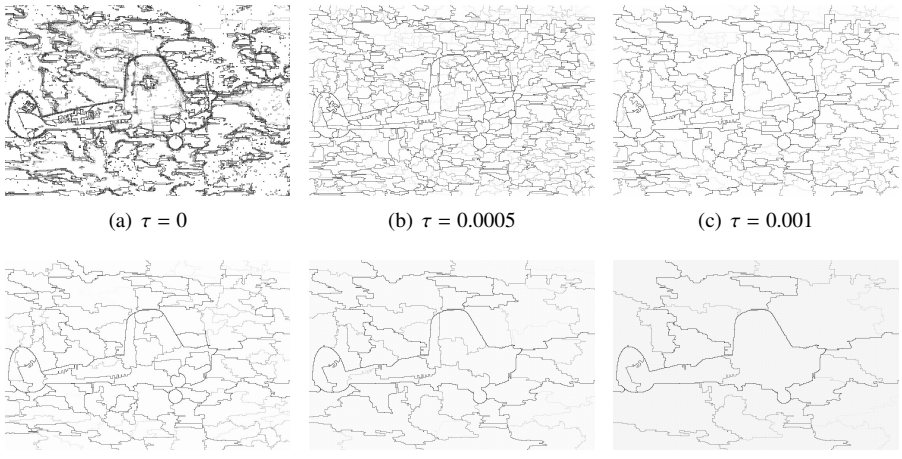
Let us also observe the role of the area-filtering post-processing. Figure 6 illustrates the saliency maps of the hierarchical segmentation results of the proposed method *p-hGB*, obtained from the original image in Fig. 1 (a) with various values of area filtering parameter. It can be observed that larger regions are eliminated when larger values of the parameter τ are considered.

4.4. Running time issues

Our algorithm was implemented in C++ and runs on a standard single CPU computer (Intel Xeon(R) CPU 2.50GHz, 32GB) under CentOS. The computation times for the results illustrated in Fig. 1 (the image size is 321×481) are 0.990 seconds for the hierarchical segmentation result obtained by *p-hGB* and 0.05 seconds for each segmentation result obtained by *p-GB* with a given observation scale. Thus, if we calculate 50 segmentation results with 50 different observation scales for *p-GB*, then the total computation time requires 2.5 seconds. Note that there are typically 500 distinct observation scales computed by the *p-hGB* method.

4.5. Quantitative analysis

We now present some quantitative results that illustrate the efficiency of our hierarchical graph-based method *hGB*, compared to the non-hierarchical original method, *GB*. For this purpose, we need to compare hierarchical (or non-hierarchical) segmentation results with the ground-truth segmentations offered in the databases mentioned in Section 4.2.



(a) $\tau = 0$ (b) $\tau = 0.0005$ (c) $\tau = 0.001$

(d) $\tau = 0.002$ (e) $\tau = 0.005$ (f) $\tau = 0.01$

Figure 6: Saliency maps of the hierarchical segmentation results of the proposed method *p-hGB*, obtained from the original image in Fig. 1 (a) with various values of area filtering parameter: $\tau = 0$ (a), 0.0005 (b), 0.001 (c), 0.002 (d), 0.005 (e) and 0.01 (f).

Such an evaluation faces the following two difficulties:

- as the observation scale varies for each method, *GB* and *hGB*, we obtain a series of segmentation results for each image of the database, so that we need to choose one result among them in order to make an evaluation;
- in practice, we need to reduce the set of observation scale values, from each of which we obtain a segmentation, as there are too many conceivable values for an exhaustive evaluation.

In the following, we first explain (1) how to select a reasonable set of observation scale values for each method, and then (2) how to choose the segmentation result among a series of results obtained due to the varied settings of observation scale value. For the purpose (2), we make an optimization with respect to segmentation quality measures, which will be also described below.

4.5.1. Selection of observation scale values

In the original method *GB*, we need to tune the observation scale K in order to obtain a segmentation result. Here, we varied K from the initial value 100 with the interval of 300 until 50 different segmentations were obtained.

Concerning our method *hGB*, once a hierarchy represented by an edge-weighted graph is obtained, *i.e.* a saliency map, all segmentations can easily be inferred using only a thresholding operation on the edge weight values, or just removing the edges with highest weight values. Indeed, each calculated edge weight corresponds to the observation scale dissimilarity between the two adjacent regions shared by the edge.

Therefore, if we remove 50 edges iteratively, we obtain a set of 50 segmentations thanks to the hierarchical structure.

4.5.2. Segmentation quality measures

In order to compare each segmentation result obtained by either *GB* or *hGB* with a ground-truth, we use three different precision-recall frameworks, which are for boundaries [21], for regions [20], and for objects and parts [27]. The first two precision-recall frameworks were used for evaluating hierarchical image segmentations in [3], and it was pointed out that measures of region quality should be also considered in addition to those of boundary quality. However, the precision-recall for regions is still limited in cases of over- or under-segmentations. Pont-Tuset and Marques then has proposed the new region-based measure by classifying regions into object and part candidates in order to adapt the case that an object consists of several parts [27].

We thus use the F-measures defined from the precision-recall for boundaries, regions, and objects and parts, denoted by F_b, F_r and F_{op}, respectively, in this article[2]. Note that the segmentation is perfect when $F_* = 1$ and quite different from the ground-truth when $F_* = 0$. In addition to those F-measures, the corresponding precision-recall curves are also used in order to capture their natures with varying observation scale values.

4.5.3. Optimal F-measures

For each pair made of an image segmentation and the associated ground-truth, one F-measure value is obtained. Thus, for one image, a series of F-measures can be obtained while the observation scale varies. In order to synthesize the whole series of F-measures, several choices can be made according to [3]. We can:

1. keep the best F-measure obtained for each image of the database; or
2. keep the F-measure for a constant scale over the database, the constant scale being chosen to maximize the average F-measure of the overall database.

They are called Optimal Image Scale (OIS) and Optimal Database Scale (ODS), respectively. Let $F(I, K)$ be the F-measure calculated from the segmentation result obtained from a given observation scale K for an image I in the dataset. Then the F-measure for the dataset with the ODS (resp. OIS) setting is obtained by $F^{ODS} = \max_K \text{avg}_I\ F(I, K)$ (resp. $F^{OIS} = \text{avg}_I\ \max_K F(I, K)$). Naturally from these definitions, F^{OIS} is always larger than F^{ODS} as the observation scale K is better chosen for each image for OIS.

4.5.4. Optimal parameter setting for pre- and post-processings

As seen in Section 4.1, both of the methods *GB* and *hGB* are accompanied with the pre-processing - Gaussian smoothing - and the post-processing - area filtering, which contain the parameters σ and τ, respectively.

[2]Other well-known measures of regions quality, such as segmentation covering [3], probabilistic Rand index [34], and variation of information [22], were also used for comparisons, whose results are shown in Appendix.

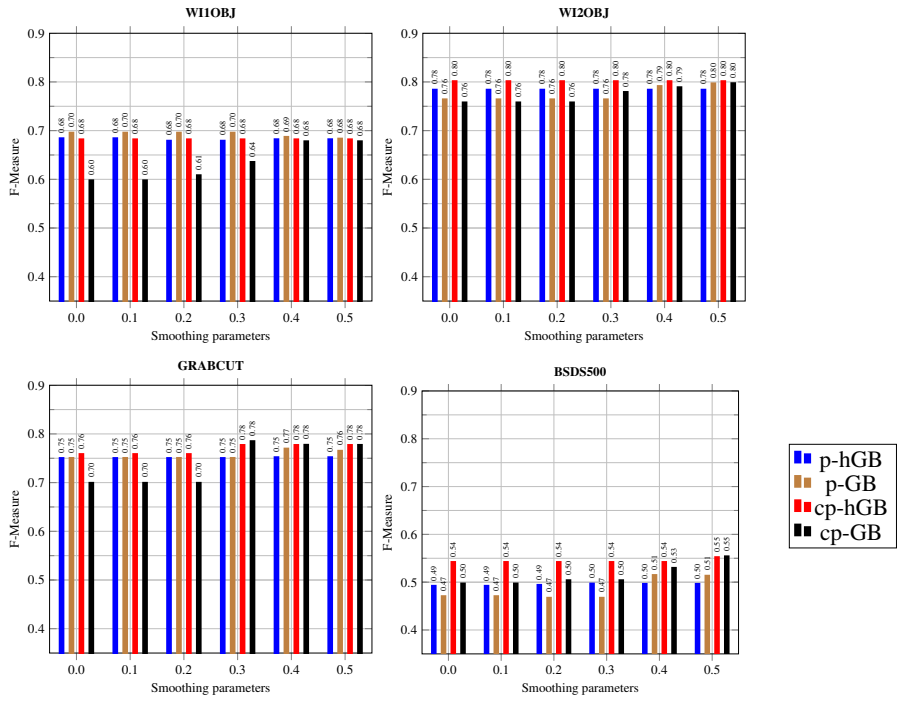


Figure 7: F_r-measure comparisons between the four segmentation results of *p-hGB*, *p-GB*, *cp-hGB* and *cp-GB* on each of the databases, **BSDS500**, **GRABCUT**, **WI1OBJ** and **WI2OBJ**, under the ODS setting with the variation of the smoothing parameter σ from 0.0 to 0.5.

The smoothing parameter σ ranges from 0 to 0.5. Concerning the area-filtering, the component size ratio with respect to the image size τ ranges from 0.001 to 0.009. In order to identify those parameter values, $9\times$ 6 runs were made for the best F-measure, F^{OIS} and F^{ODS}. In other words, we maximize the best F-measures for 9×6 value pairs $(\sigma,\tau)\in[0,0.5]\times[0.001,0.009]$.

4.5.5. Robustness under smoothing parameter variations

Changing the pre-processing smoothing parameter σ from 0 to 0.5, we observed the variations of F-measures for regions, F_r, of the *hGB* and *GB* segmentation results. Figures 7 and 8 present the optimal F_r-measures for each database under the ODS and OIS settings, respectively. They show that the results of our methods, *p-hGB* and *cp-hGB*, are robust to σ value variations for any dataset while those of *p-GB* and *cp-GB* are influenced by σ values. In general, the method *cp-hGB* provides either the best results or the ones close to the best ones. Similar results by using other region-quality measures are also given in Appendix Appendix B.

To better understand the above comparison results, we also present the distribution of the best F_r-measures with the OIS setting, illustrated in Fig. 9. Differently from Figure 8, which simply presents the average of all the best F_r-measures, each of which is computed from every segmentation result, Fig. 9 represents the distribution of the

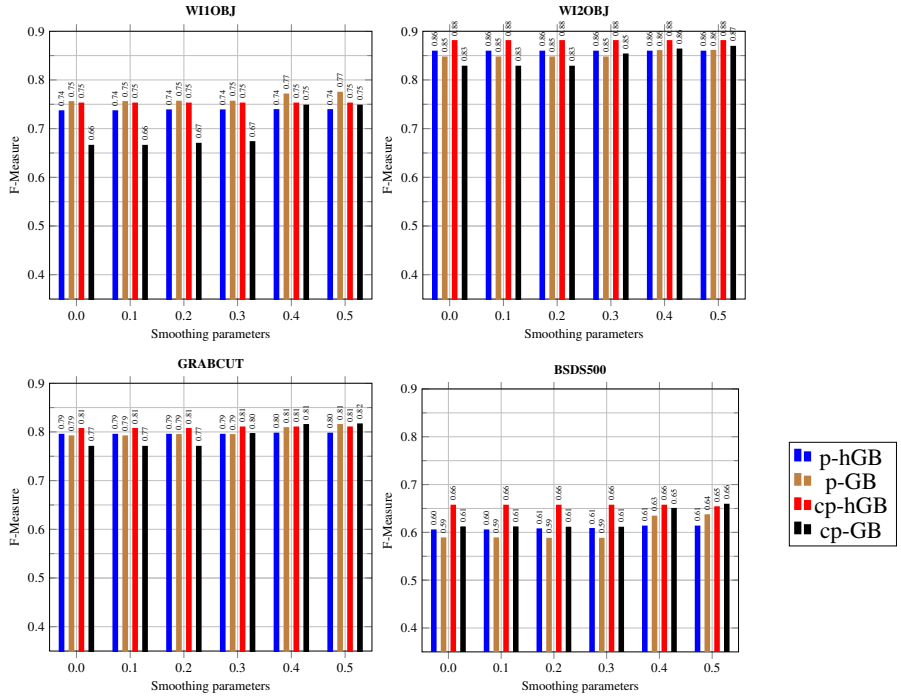


Figure 8: F_r-measure comparisons between the four segmentation results of *p-hGB*, *p-GB*, *cp-hGB* and *cp-GB* on each of the databases, **BSDS500**, **GRABCUT**, **WI1OBJ** and **WI2OBJ**, under the OIS setting with the variation of the smoothing parameter σ from 0.0 to 0.5.

best F_r-measures for individual images with the box-and-whisker plot, in which the five different values are illustrated: the median; the upper quartile; the lower quartile; the upper whisker; and the lower whisker. In other words, we can compare two distributions by using this plot, for example, between *GB* and *hGB*, or with and without smoothing. From such comparisons, we can observe that, in absence of the smoothing pre-processing, the F_r-measures of *hGB* are better than or equal to those of *GB*, as *hGB* provides higher medians than those of *GB*, except for *p-hGB* on **WI1OBJ**; even in this case, the quartile box of *p-hGB* is smaller than that of *p-GB* indeed. In presence of the smoothing pre-processing, however, it is not the case. Note that the best σ value from the parameter variations in Fig. 8 was used here.

4.5.6. Region quality evaluations

By using the F-measures for regions F_r and for objects and parts F_{op}, we made pairwise region-quality comparisons of segmentation results, as illustrated in Figs. 10 and 11, following the same presentation as in [3], in which each red point has the F-measures (with OIS) of the results of *hGB* and *GB* as its x- and y-coordinates, respectively, for each image. When no smoothing is considered, *hGB* provides either better results than or equivalent to *GB* (see Figs. 10 (a, b) and 11 (a, b)) as there are more red dots below the line $y = x$ than those above the line. Every figure also gives the

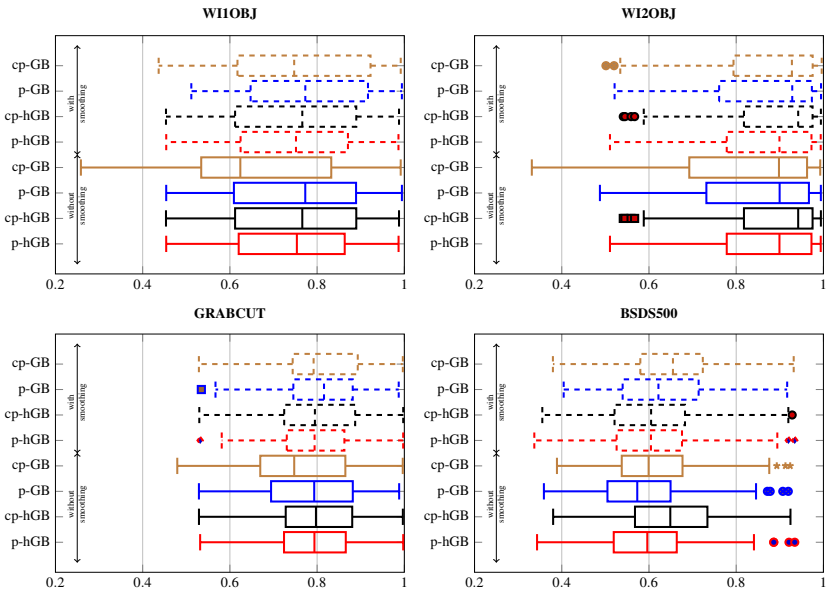


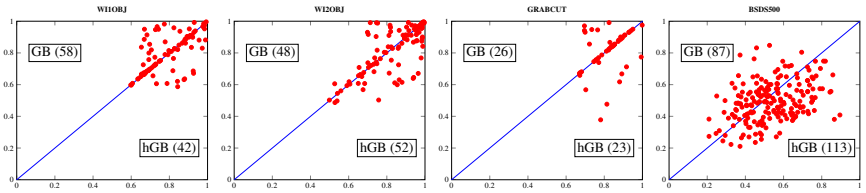
Figure 9: Box-and-whisker plots of the distribution of the best F_r-measures under the OIS setting, for the four segmentation results of *p-hGB*, *p-GB*, *cp-hGB* and *cp-GB* on each of the databases, **BSDS500**, **GRABCUT**, **WI1OBJ** and **WI2OBJ**, with and without the smoothing pre-processing.

number of red points in both sides; for example, in Fig. 10 (a-rightmost) there are 113 images of the database **BSDS500** whose F_r-measures obtained from *p-hGB* is better than those obtained from *p-GB*. When smoothing is considered, *hGB* provides almost equivalent results to *GB* (see Figs. 10 (c, d) and 11 (c, d)).

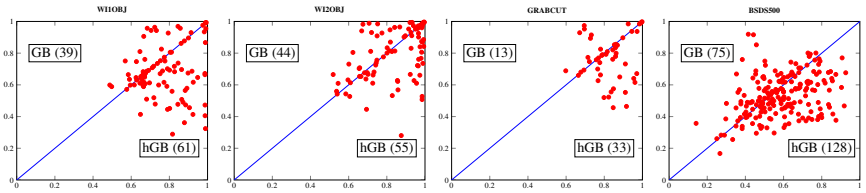
This observation is also confirmed by Table 1, in which the best method was chosen between *hGB* and *GB*, with 5% of confidence interval, from each pairwise region-quality comparison for every database in Figs. 10 and 11. The interpretation is made as follows; if the confidence interval contains zero, then both methods are chosen as they are considered to be equivalent; otherwise, the confidence interval allows us to choose the best method: if the interval is completely included in the positive (resp. negative) side, then *hGB* (resp. *GB*) is chosen to be the best. As seen from Table 1, *cp-hGB* (or both) is always better than or equivalent to *cp-GB* when no smoothing is applied, and when smoothing is applied, both are equivalent. Comparing between *p-hGB* and *p-GB*, when there is no smoothing, *p-hGB* is either better or equivalent, and vice-versa when there is smoothing.

4.5.7. Evaluations based on SESIM framework

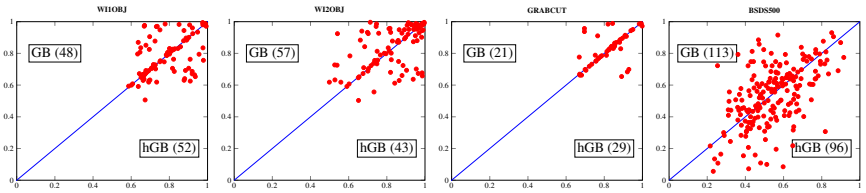
In this section, we consider the precision-recall curves for boundaries, regions and objects and parts, whose F-measures correspond to F_b, F_r and F_{op}, respectively, for comparing our segmentation results of *hGB* with those of *GB* in order to capture their dynamics with varying observation scale values. We made use of the software pack-



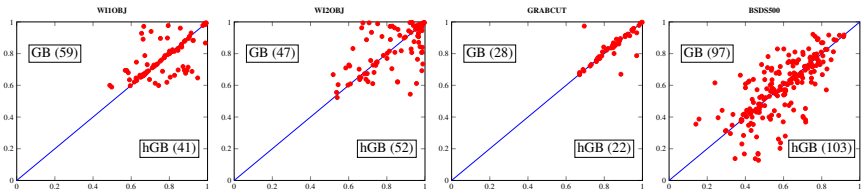
(a) type graph (p) and smoothing (no)



(b) type graph (cp) and smoothing (no)



(c) type graph (p) and smoothing (yes)

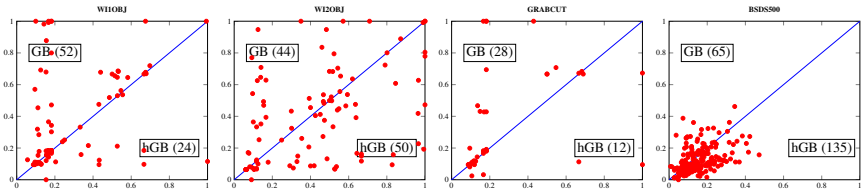


(d) type graph (cp) and smoothing (yes)

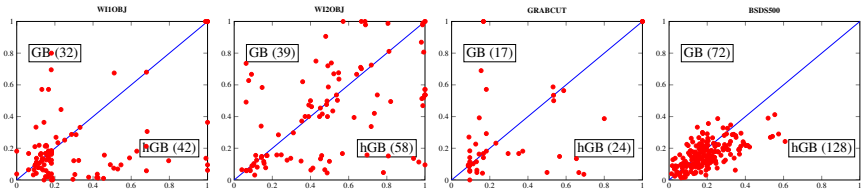
Figure 10: Pairwise region-quality comparison of segmentation results, in which each red point has the F_r-measures (with OIS) of the results of *hGB* and *GB* as its x- and y-coordinates for each image. Two different graph types, p and cp, are considered with and without smoothing pre-processing.

age SEISM [27], with which we can compare our results with those of not only *p-GB* but also the following state-of-the-art methods on the database **BSDS500**: Ultrametric Contour Maps (UCM) [3], Normalized Cuts (NCuts) [30], Mean Shift (MShift) [6], NWMC Binary Partition Tree (NWMC) [36], and IID-KL Binary Partition Tree (IID-KL) [4]. In fact, we added our segmentation results of *p-hGB* and *cp-hGB* to the others that were already pre-computed [27], as shown in Figure 12. Obviously, it is particularly interesting to compare our method *p-hGB* (and *cp-hGB*) against *p-GB* which considers the same dissimilarity measure in the figures.

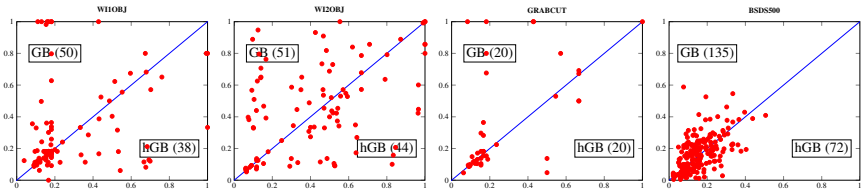
Figure 12 shows that, depending on chosen F-measures, the curves drawn from the same segmentation results are different. In addition, the curves for objects and parts



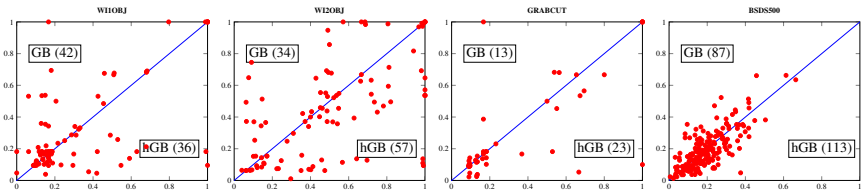
(a) type graph (p) and smoothing (no)



(b) type graph (cp) and smoothing (no)



(c) type graph (p) and smoothing (yes)



(d) type graph (cp) and smoothing (yes)

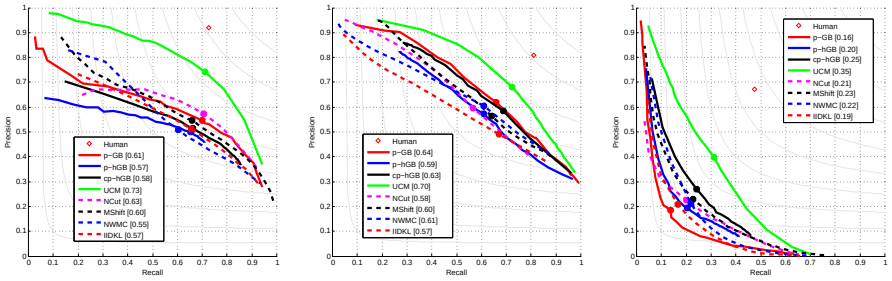
Figure 11: Pairwise region-quality comparison of segmentation results, in which each red point has the F_{op}-measures (with OIS) of the results of *hGB* and *GB* as its x- and y-coordinates for each image. Two different graph types, p and cp, are considered with and without smoothing pre-processing.

indicate that *cp-hGB* provides better results than *p-GB*, while those for boundaries and regions indicate that *p-GB* provides better results than *cp-hGB*.

Some examples of segmentation results are illustrated in Figs. 13 and 14; given the original image and the ground truth (the average among 5 human segmentations) illustrated in (a) and (b), the best segmentation results obtained by *p-GB*, *p-hGB* and *cp-hGB* with respect to F_{op} in Fig. 12 are shown in (c), (d) and (e), respectively. The results in Figs. 13 (c) and 14 (c) consist of thin long regions that are located around the object boundaries, but do not enclose the objects. Thus they are visually poorer results than those in Figs. 13 (d, e) and 14 (d, e), while the F_b and F_r values are similar between (c) and (d, e) in Figs. 13 and 14 (or (c) have even higher values than (d, e)). A

Database	Methods	Smoothing	F-measure for regions Confidence interval (5%)	F-measure for regions The best method	F-measure for object and parts Confidence interval (5%)	F-measure for object and parts The best method
BSDS500	p-hGB and p-GB	no	[0.006,0.031]	p-hGB	[0.017,0.042]	p-hGB
		yes	[-0.033,-0.013]	p-GB	[-0.023,0.004]	Equivalent
	cp-hGB and cp-GB	no	[0.033,0.057]	cp-hGB	[0.024,0.048]	cp-hGB
		yes	[-0.012,0.002]	Equivalent	[-0.008,0.014]	Equivalent
GRABCUT	p-hGB and p-GB	no	[-0.026,0.028]	Equivalent	[-0.147,0.017]	Equivalent
		yes	[-0.032,0.002]	Equivalent	[-0.139,0.004]	Equivalent
	cp-hGB and cp-GB	no	[0.011,0.070]	cp-hGB	[-0.064,0.092]	Equivalent
		yes	[-0.014,0.013]	Equivalent	[-0.025,0.094]	Equivalent
WEIZMANN 1 OBJ	p-hGB and p-GB	no	[-0.045,0.009]	Equivalent	[-0.132,-0.021]	p-GB
		yes	[-0.055,-0.015]	p-GB	[-0.096,0.024]	Equivalent
	cp-hGB and cp-GB	no	[0.061,0.130]	cp-hGB	[0.045,0.150]	cp-hGB
		yes	[-0.013,0.022]	Equivalent	[-0.038,0.049]	Equivalent
WEIZMANN 2 OBJ	p-hGB and p-GB	no	[-0.006,0.031]	Equivalent	[-0.033,0.090]	Equivalent
		yes	[-0.019,0.019]	Equivalent	[-0.085,0.035]	Equivalent
	cp-hGB and cp-GB	no	[0.029,0.079]	cp-hGB	[0.019,0.139]	cp-hGB
		yes	[-0.006,0.030]	Equivalent	[0.005,0.115]	cp-hGB

Table 1: Best method choice with confidence interval for each pairwise comparison in Figs. 10 (with F_r-measures) and 11 (with F_{op}-measures).



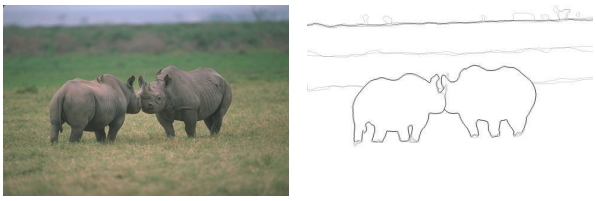
(a) boundaries (F_b) (b) regions (F_r) (c) objects and parts (F_{op})

Figure 12: Precision-recall curves for boundaries (a), for regions (b) and for objects and parts (c). The curves were drawn from points plotted by the segmentation results obtained by the six state-of-the-art segmentation methods (UCM, *p-GB* NWMC, IID-KL, MShift and NCuts) thanks to the software SEISM [27] and the proposed methods (*p-hGB* and *cp-hGB*). The marker on each curve is placed on the Optimal Dataset Scale (ODS). The isolated red diamond refer to the human performance assessed.In the legend, the F-measures of the marked points on each curve is presented in brackets.

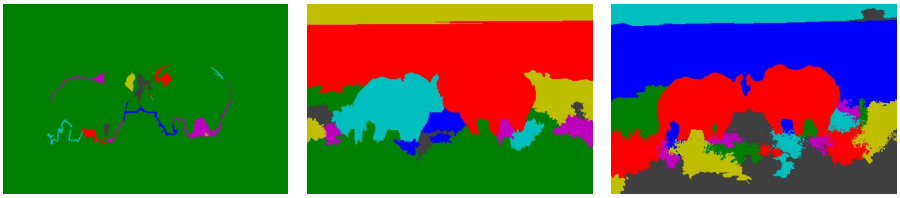
similar observation was already made in [27], and this indicates that F_{op} can evaluate segmentation results differently from F_b and F_r. It should be also noticed that F_{op} gives relatively smaller values than F_b and F_r in general.

5. Conclusions

In this paper, we applied the toolbox proposed in [10] to develop a hierarchical version of some graph-based image segmentation algorithms relying on region dissimilarity. The main example of such a criterion is the one proposed in [12], and we performed an extensive set of experiments that demonstrates that our algorithm achieves result on par with the one of [12], with the significant benefit of being much easier to tune. Indeed, the user can just select the level in the hierarchy, controlling the desired



(a) original (b) ground truth



(c) result by *p-GB*: $F_r = 0.64$, $F_b = 0.62$, $F_{op} = 0.05$ (d) result by *p-hGB*: $F_r = 0.58$, $F_b = 0.57$, $F_{op} = 0.28$ (e) result by *cp-hGB*: $F_r = 0.49$, $F_b = 0.48$, $F_{op} = 0.20$

Figure 13: Example of segmentation results of the original image (a) obtained by *p-GB* (c), *p-hGB* (d) and *cp-hGB* (e). The F-measures, illustrated on legends, for regions (F_r), boundaries (F_b) and object and parts (F_{op}) were based on the ground truth illustrated in (b) as the saliency map.

(a) original (b) ground truth

(c) result by *p-GB*: $F_r = 0.77$, $F_b = 0.71$, $F_{op} = 0.18$ (d) result by *p-hGB*: $F_r = 0.60$, $F_b = 0.69$, $F_{op} = 0.36$ (e) result by *cp-hGB*: $F_r = 0.69$, $F_b = 0.71$, $F_{op} = 0.50$

Figure 14: Example of segmentation results of the original image (a) obtained by *p-GB* (c), *p-hGB* (d) and *cp-hGB* (e). The F-measures, illustrated on legends, for regions (F_r), boundaries (F_b) and object and parts (F_{op}) were based on the ground truth illustrated in (b) as the saliency map.

number of regions.

We believe such results are an incentive to continue developing the generic tools proposed in [10]. Many other criteria, such as the one proposed in [25], can be included in our framework [16], despite being significantly more complex. Other applications, such as dealing with video, are also possible, and our approach can provide state-of-the-art results [32]. We also envision extending such approaches for classification problems (*i.e.* non image data), as proposed in [10]. Last, but not the least considering the current trend in computer vision, an interesting perspective is obviously, on a specific application, to use learning techniques and train a criterion to choose the correct region. First results in that direction are encouraging [7].

For all these research directions, the question of evaluation is a fundamental one. In this paper, we used the existing ground-truth segmentations available in the literature. One should note that such segmentations are not hierarchical, and as such, we can not truly assess the benefit of a hierarchical organisation of hierarchical approaches. A considerable amount of work is nowadays devoted to the construction of a sound evaluation framework [3, 26, 27, 28]. Although the question of the evaluation of hierarchies is an even more complex question, we are deeply convinced that the computer-vision community at large would benefit if hierarchical ground truths were available.

On a more theoretical level, we would like to make a formal study of the various algorithms that can transform a hierarchy, in order to obtain a better efficiency: on the one hand, we believe some speed improvements are possible with respect to what we are doing today. On the other hand, it would be nice to guarantee some structural properties on the resulting hierarchy of segmentations, such as a "not-too-coarse" one.

Acknowledgements

The research leading to these results has received funding from the French Agence National de la Recherche (contract ANR-2010-BLAN-0205-03), the French Committee for the Evaluation of Academic and Scientific Cooperation with Brazil, and the Brazilian Federal Agency of Support and Evaluation of Postgraduate Education (program CAPES/PVE: grant 064965/2014-01, and program CAPES/COFECUB: grant 592/08).

Appendix A. Hierarchical area filtering

As the proposed segmentation method provides a hierarchical graph-based result represented by a spanning tree (see Section 3), we adapt our area-filtering for such hierarchical outputs as follows. Area-filtering post-processing eliminates connected components whose size is smaller than a given value M. This can be realized by re-weighting the spanning tree that is a segmentation result obtained by Algorithm 1. The algorithm is shown in Algorithm 2, which simply replaces the edge weight with zero if the size of one of connected components merged by this edge is smaller than a given value M.

Algorithm 2: Hierarchical area-filtering

Data: A minimum spanning tree $T = (V, E')$ of f
Data: A minimum area size M
Result: A map f' from E' to $\mathbb{R}^+$

for *each* $u = \{x, y\} \in E'$ *in non-decreasing order for* f **do**
  if $|\mathbf{C}(f_\lambda^V(T))_x| \geq M$ *and* $|\mathbf{C}(f_\lambda^V(T))_y| \geq M$ **then** $f'(u) := f(u)$;
  else $f'(u) := 0$;

Appendix B. Robustness under smoothing parameter variations with other measures

Changing the pre-processing smoothing parameter σ from 0 to 0.5, we observed the variations of the segmentation covering measures of the *hGB* and *GB* segmentation results. Figure B.15 and B.16 present the optimal measure for each database under ODS and OIS, respectively.

Appendix C. Region quality evaluations with other measures

Table C.2 shows some performance measures related to the segmentation task (segmentation covering, probabilistic Rand index, and variation of information). Here, it is also possible to observe the robustness with respect to smoothing operation.

References

[1] Alpert, S., Galun, M., Basri, R., Brandt, A.: Image segmentation by probabilistic bottom-up aggregation and cue integration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2007)

[2] Alpert, S., Galun, M., Brandt, A., Basri, R.: Image segmentation by probabilistic bottom-up aggregation and cue integration. IEEE Trans. Pattern Anal. Mach. Intell. **34**(2), 315–327 (2012)

Database	Graph	Method	Smoothing	Covering			PRI		VI	
				ODS	OIS	Best	ODS	OIS	ODS	OIS
GRABCUT	p	hGB	no	0.71	0.77	0.81	0.71	0.78	0.69	0.65
			yes	0.71	0.77	0.81	0.71	0.79	0.68	0.65
	p	GB	no	0.73	0.78	0.82	0.73	0.77	0.76	0.72
			yes	0.74	0.79	0.83	0.74	0.79	0.68	0.62
	cp	hGB	no	0.73	0.79	0.83	0.73	0.80	0.72	0.67
			yes	0.74	0.79	0.83	0.75	0.80	0.67	0.64
	cp	GB	no	0.69	0.77	0.81	0.69	0.76	0.99	0.86
			yes	0.74	0.80	0.83	0.74	0.80	0.66	0.63
WEIZMANN1OBJ	p	hGB	no	0.65	0.72	0.76	0.67	0.78	0.81	0.75
			yes	0.65	0.72	0.76	0.66	0.78	0.80	0.74
	p	GB	no	0.67	0.75	0.78	0.68	0.77	0.82	0.76
			yes	0.67	0.76	0.79	0.69	0.79	0.82	0.71
	cp	hGB	no	0.65	0.74	0.77	0.68	0.78	0.81	0.73
			yes	0.65	0.74	0.77	0.68	0.78	0.81	0.73
	cp	GB	no	0.60	0.68	0.71	0.62	0.70	1.34	1.21
			yes	0.67	0.74	0.78	0.69	0.78	0.87	0.77
WEIZMANN2OBJ	p	hGB	no	0.76	0.86	0.88	0.77	0.88	0.74	0.59
			yes	0.76	0.86	0.88	0.77	0.88	0.74	0.59
	p	GB	no	0.75	0.84	0.87	0.75	0.85	0.78	0.61
			yes	0.79	0.86	0.88	0.79	0.87	0.73	0.56
	cp	hGB	no	0.80	0.88	0.90	0.80	0.89	0.72	0.51
			yes	0.80	0.88	0.90	0.80	0.89	0.72	0.51
	cp	GB	no	0.75	0.84	0.86	0.75	0.84	0.84	0.70
			yes	0.79	0.87	0.89	0.80	0.88	0.73	0.56
BSDS500	p	hGB	no	0.45	0.54	0.63	0.76	0.81	2.30	1.83
			yes	0.45	0.55	0.63	0.77	0.81	2.28	1.80
	p	GB	no	0.43	0.53	0.68	0.75	0.80	2.33	1.91
			yes	0.46	0.58	0.68	0.75	0.82	2.19	1.74
	cp	hGB	no	0.50	0.60	0.68	0.77	0.83	2.08	1.65
			yes	0.51	0.59	0.68	0.78	0.83	2.03	1.66
	cp	GB	no	0.45	0.56	0.68	0.75	0.81	2.17	1.80
			yes	0.50	0.60	0.69	0.78	0.83	2.06	1.63

Table C.2: Performances of our method and the method GB [12] using three different measures: Ground-truth Covering (GT Covering) , Variation Information and Probabilistic Rand Index. In both case, the input images are and are not submitted to smoothing operation. The presented scores are optimal considering a constant scale parameter for the whole dataset (ODS) and a scale parameter varying for each image (OIS). See [3] for more details on the evaluation method.

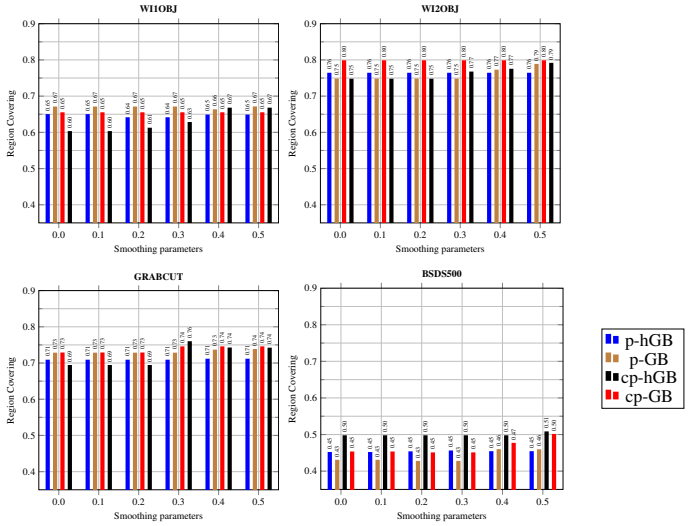


Figure B.15: Evaluation of segmentation algorithms on four databases using GT region covering for optimal scale for the entire dataset (ODS) in order to understand the their behaviour when the images are submitted to smoothing operations (σ varying from 0.0 to 0.5).

[3] Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**, 898–916 (2011). DOI http://doi.ieeecomputersociety.org/10.1109/TPAMI.2010.161

[4] Calderero, F., Marques, F.: Region merging techniques using information theory statistical measures. Trans. Img. Proc. **19**(6), 1567–1586 (2010). DOI 10.1109/TIP.2010.2043008. URL `http://dx.doi.org/10.1109/TIP.2010.2043008`

[5] Cardelino, J., Caselles, V., Bertalmio, M., Randall, G.: A contrario selection of optimal partitions for image segmentation. SIAM Journal on Imaging Sciences **6**(3), 1274–1317 (2013)

[6] Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 603–619 (2002). DOI 10.1109/34.1000236. URL `http://dx.doi.org/10.1109/34.1000236`

[7] Couprie, C., Farabet, C., Najman, L., Lecun, Y.: Convolutional nets and watershed cuts for real-time semantic labeling of rgbd video. The Journal of Machine Learning Research **15**, 3489–3511 (2014)

[8] Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed cuts: Minimum spanning forests and the drop of water principle. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(8), 1362–1374 (2009)

[9] Cousty, J., Najman, L., Kenmochi, Y., Guimarães, S.: New characterizations of minimum spanning trees and of saliency maps based on quasi-flat zones. In:

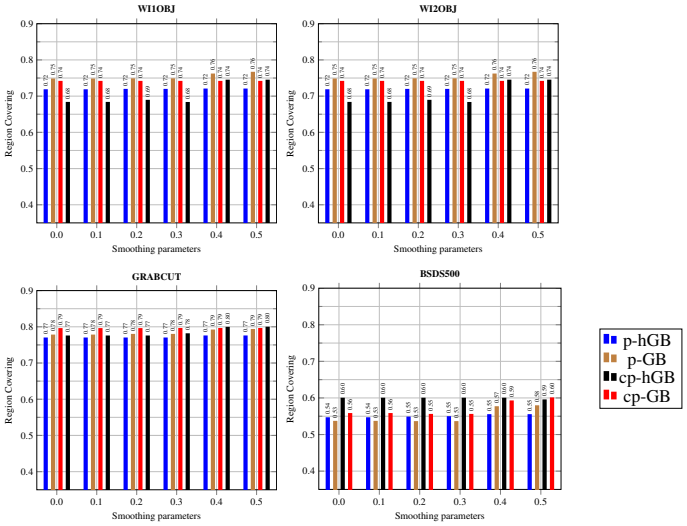


Figure B.16: Evaluation of segmentation algorithms on four databases using GT region covering for optimal scale for each image (OIS) in order to understand the their behaviour when the images are submitted to smoothing operations (σ varying from 0.0 to 0.5).

Mathematical Morphology and Its Applications to Signal and Image Processing, pp. 205–216. Springer (2015)

[10] Cousty, J., Najman, L., Kenmochi, Y., Guimarães, S.: Hierarchical segmentations with graphs: quasi-flat zones, minimum spanning trees, and saliency maps. Technical report, Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge (2016). URL `https://hal.archives-ouvertes.fr/hal-01344727`

[11] Cousty, J., Najman, L., Perret, B.: Constructive links between some morphological hierarchies on edge-weighted graphs. In: Proceedings of 11th International Symposium on Mathematical Morphology - ISMM 2013, *Lecture Notes in Computer Science*, vol. 7883, pp. 86–97. Springer (2013)

[12] Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. International Journal of Computer Vision **59**, 167–181 (2004)

[13] Guigues, L., Cocquerez, J.P., Men, H.L.: Scale-sets image analysis. International Journal of Computer Vision **68**(3), 289–317 (2006)

[14] Guigues, L., Le Men, H., Cocquerez, J.P.: The hierarchy of the cocoons of a graph and its application to image segmentation. Pattern Recognition Letters **24**(8), 1059–1066 (2003)

[15] Guimarães, S.J.F., Cousty, J., Kenmochi, Y., Najman, L.: A hierarchical image segmentation algorithm based on an observation scale. In: SSPR/SPR, pp. 116–125 (2012)

[16] Guimarães, S.J.F., do Patrocínio Jr., Z.K.G., Kenmochi, Y., Cousty, J., Najman, L.: Hierarchical image segmentation relying on a likelihood ratio test. In: V. Murino, E. Puppo (eds.) Image Analysis and Processing - ICIAP 2015 - 18th International Conference, Genoa, Italy, September 7-11, 2015, Proceedings, Part II, *Lecture Notes in Computer Science*, vol. 9280, pp. 25–35. Springer (2015). DOI 10.1007/978-3-319-23234-8_3. URL http://dx.doi.org/10.1007/978-3-319-23234-8_3

[17] Haxhimusa, Y., Kropatsch, W.: Hierarchy of partitions with dual graph contraction. In: Pattern Recognition, pp. 338–345. Springer (2003)

[18] Haxhimusa, Y., Kropatsch, W.G.: Segmentation graph hierarchies. In: A.L.N. Fred, T. Caelli, R.P.W. Duin, A.C. Campilho, D. de Ridder (eds.) Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshops, SSPR 2004 and SPR 2004, Lisbon, Portugal, August 18-20, 2004 Proceedings, *Lecture Notes in Computer Science*, vol. 3138, pp. 343–351. Springer (2004). DOI 10.1007/978-3-540-27868-9_36. URL http://dx.doi.org/10.1007/978-3-540-27868-9_36

[19] Kiran, B.R., Serra, J.: Global–local optimizations by hierarchical cuts and climbing energies. Pattern Recognition **47**(1), 12–24 (2014)

[20] Martin, D.R.: An empirical approach to grouping and segmentation. Ph.D. thesis, EECS Department, University of California, Berkeley (2003). URL http://www.eecs.berkeley.edu/Pubs/TechRpts/2003/5252.html

[21] Martin, D.R., Fowlkes, C.C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Trans. Pattern Anal. Mach. Intell. **26**(5), 530–549 (2004). DOI 10.1109/TPAMI.2004.1273918. URL http://dx.doi.org/10.1109/TPAMI.2004.1273918

[22] Meilă, M.: Comparing clusterings: An axiomatic view. In: Proceedings of the 22Nd International Conference on Machine Learning, ICML '05, pp. 577–584. ACM, New York, NY, USA (2005). DOI 10.1145/1102351.1102424. URL http://doi.acm.org/10.1145/1102351.1102424

[23] Morris, O.J., Lee, M.J., Constantinides, A.G.: Graph theory for image analysis: an approach based on the shortest spanning tree. Communications, Radar and Signal Processing, IEE Proceedings F **133**(2), 146 –152 (1986)

[24] Najman, L.: On the equivalence between hierarchical segmentations and ultrametric watersheds. Journal of Mathematical Imaging and Vision **40**, 231–247 (2011)

[25] Nock, R., Nielsen, F.: Statistical region merging. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(11), 1452–1458 (2004)

[26] Perret, B., Cousty, J., Ura, J.C.R., Guimarães, S.J.F.: Evaluation of morphological hierarchies for supervised segmentation. In: Mathematical Morphology and Its

Applications to Signal and Image Processing, pp. 39–50. Springer International Publishing (2015)

[27] Pont-Tuset, J., Marqués, F.: Measures and meta-measures for the supervised evaluation of image segmentation. In: Computer Vision and Pattern Recognition (CVPR) (2013)

[28] Pont-Tuset, J., Marques, F.: Supervised evaluation of image segmentation and object proposal techniques. IEEE Transactions on Pattern Analysis and Machine Intelligence (2015). To appear

[29] Rother, C., Kolmogorov, V., Blake, A.: "grabcut": Interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. **23**(3), 309–314 (2004). DOI 10.1145/1015706.1015720. URL http://doi.acm.org/10.1145/1015706.1015720

[30] Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000). DOI 10.1109/34.868688. URL http://dx.doi.org/10.1109/34.868688

[31] Soille, P.: Constrained connectivity for hierarchical image partitioning and simplification. Pattern Analysis and Machine Intelligence, IEEE Transactions on **30**(7), 1132–1145 (2008)

[32] de Souza, K.J.F., de Albuquerque Araújo, A., do Patrocínio Jr., Z.K.G., Guimarães, S.J.F.: Graph-based hierarchical video segmentation based on a simple dissimilarity measure. Pattern Recognition Letters **47**, 85–92 (2014). DOI 10.1016/j.patrec.2014.02.016. URL http://dx.doi.org/10.1016/j.patrec.2014.02.016

[33] Tarjan, R.E.: Efficiency of a good but not linear set union algorithm. Journal of the ACM **22**(2), 215–225 (1975)

[34] Unnikrishnan, R., Pantofaru, C., Hebert, M.: Toward objective evaluation of image segmentation algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 929–944 (2007). DOI 10.1109/TPAMI.2007.1046. URL http://dx.doi.org/10.1109/TPAMI.2007.1046

[35] Varas, D., Alfaro, M., Marqués, F.: Multiresolution hierarchy co-clustering for semantic segmentation in sequences with small variations. In: ICCV - International Conference on Computer Vision (2015). URL http://arxiv.org/abs/1510.04842

[36] Vilaplana, V., Marques, F., Salembier, P.: Binary partition trees for object detection. Image Processing, IEEE Transactions on **17**(11), 2201–2216 (2008). DOI 10.1109/TIP.2008.2002841

[37] Xu, Y., Géraud, T., Najman, L.: Connected filtering on tree-based shape-spaces. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(6), 1126–1140 (2016)

[38] Zahn, C.T.: Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Trans. Comput. **20**, 68–86 (1971)