HAL
open science

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

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

To cite this version:

Silvio J. 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. Mathematical Morphology - Theory and Applications, 2017, 10.1515/mathm-2017-0004 . hal-01342967v4

HAL Id: hal-01342967

https://hal.science/hal-01342967v4

Submitted on 24 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Research Article

Silvio Guimarães*, Yukiko Kenmochi, Jean Cousty, Zenilton Patrocinio Jr., and Laurent Najman

Hierarchizing graph-based image segmentation algorithms relying on region dissimilarity

the case of the Felzenszwalb-Huttenlocher method

Abstract: This article is a first attempt towards a general theory for hierarchizing non-hierarchical image segmentation method depending on a region-dissimilarity parameter which controls the desired level of simplification: each level of the hierarchy is "as close as possible" to the result that one would obtain with the non-hierarchical method using the corresponding scale as simplification parameter. The introduction of this hierarchization problem in the form of an optimization problem, as well as the proposed tools to tackle it, is an important contribution of the present article. Indeed, with the hierarchized version of a segmentation method, the user can just select the level in the hierarchy, controlling the desired number of regions or can leverage on any of the tools introduced in hierarchical analysis. The main example investigated in this study is the criterion proposed by Felzenszwalb and Huttenlocher for which we show that the results of the hierarchized version of the segmentation method are better than those of the original one with the added property that it satisfies the strong causality and location principles from scale-sets image analysis. An interesting perspective of this work, considering the current trend in computer vision, is obviously, on a specific application, to use learning techniques and train a criterion to choose the correct region.

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. The literature 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 [1], [35], [38]).

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.* [10], a hierarchy satisfies two important principles of multi-scale image analysis.

***Corresponding Author: Silvio Guimarães:** PUC Minas - ICEI - DCC - VIPLAB, E-mail: sjamil@pucminas.br
Yukiko Kenmochi: Université Paris-Est, LIGM, CNRS, E-mail: yukiko.kenmochi@esiee.fr
Jean Cousty: Université Paris-Est, LIGM, ESIEE Paris, E-mail: jean.cousty@esiee.fr
Zenilton Patrocinio Jr.: PUC Minas - ICEI - DCC - VIPLAB, E-mail: zenilton@pucminas.br
Laurent Najman: Université Paris-Est, LIGM, ESIEE Paris, E-mail: laurent.najman@esiee.fr

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

Since the early work of [28], hierarchical image analysis has been the subject of intense research. For instance, one can refer to hierarchical watersheds, pioneered in [2, 18, 22, 25], to quasi-flat zones hierarchies, studied notably in [23], to binary partition trees, introduced in [33], and to the scale-set theory, initiated in [10]. In the few last years, hierarchical segmentation has become a hot topic as attested by the popularity of [1], which presents a hierarchical segmentation machinery that reaches excellent practical results on the Berkeley image segmentation dataset.

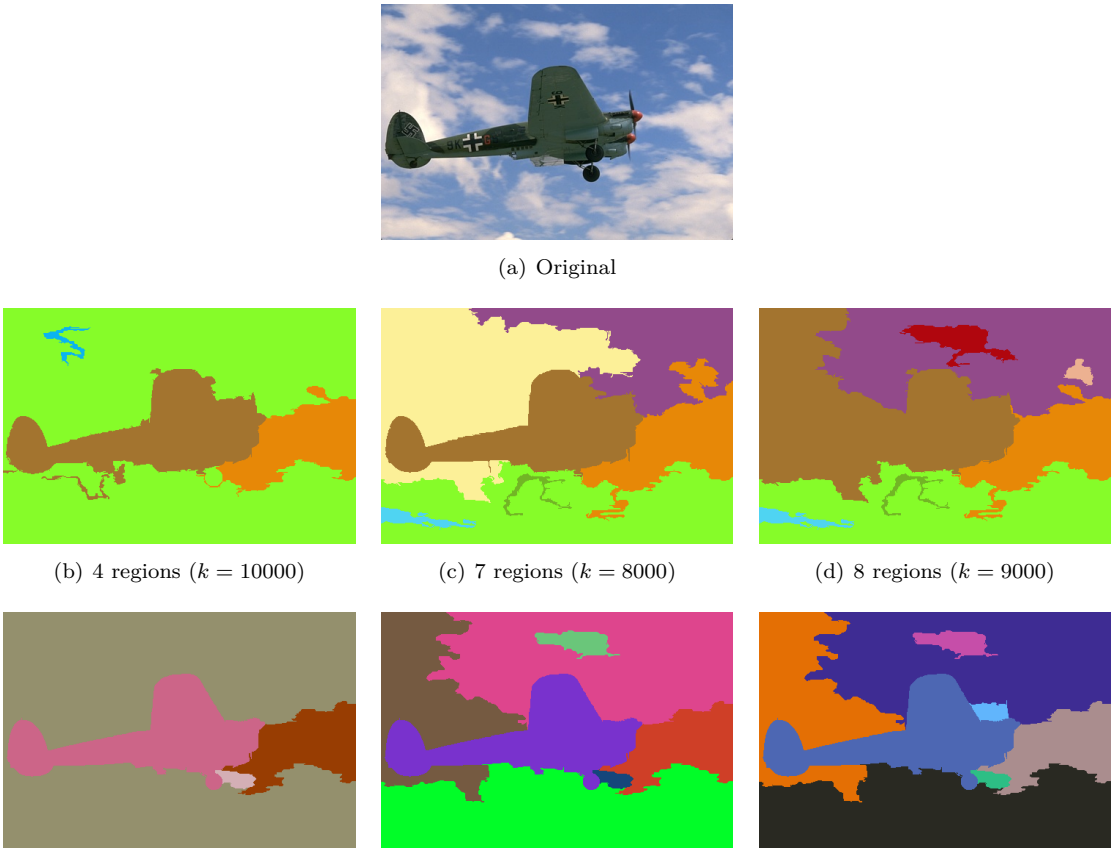
On the other hand, any hierarchy can be represented with a tree, specifically with a minimum spanning tree (MST). The first appearance of this tree in pattern recognition dates back to the seminal work of Zahn [40]. Lately, its use for image segmentation was introduced by Morris *et al.* [24] in 1986. In 2004, both Felzenszwalb and Huttenlocher [9] and Nock and Nielsen [27] proposed an image segmentation method in which the pixel-merging order is similar to the creation of an MST, so-called "scale of observation". The methods, while being very effective in its own right, do 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 of the Felzenszwalb-Huttenlocher method [9] is demonstrated in Fig. 1 (b-d).
– 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 generated by the Felzenszwalb-Huttenlocher method [9] are also illustrated in Fig. 1 (b-d).

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.* [10], which find optimal cuts in the first hierarchy. This work has been extended in several directions, see for example [3] and [17]. It is shown in [26] that mathematical morphology provides tools and operators to modify hierarchies, in a spirit similar to what is achieved in [10]. In fact, any hierarchy can be seen as a graph on which we can apply any graph-based operator [39], and the well-known watershed operator is itself related to an MST [5]. However, such approaches can not deal with a criterion such as the ones proposed in [9, 27] for the merging of regions. It should be also mentioned that several authors [11, 14–16] proposed to build a hierarchy based on both internal and external contrast measures which are similar to the one proposed in [9] specifically. A survey of such hierarchical image segmentation methods can be found in [19].

In order to go beyond, we explored in [6, 7] the links between mathematical morphology and hierarchical classification. In [7], 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 article an efficient methodology for hierarchizing some image segmentation methods that rely on a dissimilarity criterion[†] The main example that will be studied in the sequel of this article is the one proposed by Felzenszwalb and Huttenlocher [9]. Specifically, using the terminology from [9] recalled in the sequel of the article, we try to select, at each level of the novel hierarchy, the largest not-too-coarse segmentation of the cor-

† This article is an extension of [12] that proposes the original idea to hierarchize the Felzenszwalb-Huttenlocher method for graph-based image segmentation [9], without using explicitly the theoretical framework of [7]. The new contributions of this article are: a formulation of the problem as an optimization in the lattice of the hierarchies and an experimental evaluation of the proposed method.



(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

Fig. 1. Examples illustrating the results of the method proposed by [9] and our method obtained from the original image (a): three image segmentations, shown in (b), (c) and (d), are obtained by the method proposed in [9] when the observation scale k is set such that $k = 10000,\ 8000,\ 9000$ that lead to 4, 7 and 8 regions, respectively. 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 illustrate 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 by removing the edges with highest weight values until we obtain the desired number of regions: both causality and location principles are respected.

responding 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 [9], 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 [9], the number of regions is decreasing when the scale parameter increases, and the contours do not move from one scale to another, as illustrated in Fig. 1 (e-g). This greatly facilitates the selection of a given partition adapted to the application under scrutiny. As the toolbox presented in [7] allows us to manipulate a hierarchy, we apply it in this article to deal with more complex region-merging criteria, the main example being the one of [9]. In other words, this is a practical example of direct use of such theoretical toolbox. Thanks to this, we can also show experimentally the differences between the segmentation results of the original and the hierarchized methods, together with their running times.

This article is organized as follows. In Section 2, based on the theoretical framework proposed in [7], 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 criteria can be

used in our proposal, but, in order to clarify and exemplify our algorithm, we use the region dissimilarity proposed in [9]. Using this criterion allows us to perform an experimental study, that is described in Sections 4 and 5. The main conclusion that can be drawn from this study is that the proposed approach performs better than the one of [9], with the significant advantage of being much easier to tune. We finally propose some further research direction in the conclusion (Section 6).

2 Basic notions

In this section, necessary notions for hierarchical graph-based segmentations are presented, together with properties on which our algorithm is based. For this purpose, we follow the presentation given in [7].

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 $\mathcal{H} = (\mathbf{P}_0, \ldots, \mathbf{P}_\ell)$ of partitions of V, 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 $(x_0, \ldots, x_\ell)$ 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 let 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 let 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 $w_\lambda^V(X)$ of X such that

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

Then, the connected-components partition $\mathbf{C}(w_\lambda^V(X))$ induced by the λ-level 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 $w_{\lambda_1}^V(X)$ is also an edge of $w_{\lambda_2}^V(X)$. Thus, any connected component of $w_{\lambda_1}^V(X)$ is included in a connected component of $w_{\lambda_2}^V(X)$. In other words, $\mathbf{C}(w_{\lambda_1}^V(X)) \preceq \mathbf{C}(w_{\lambda_2}^V(X))$ for $\lambda_1 \leq \lambda_2$. Hence, the sequence of all λ-level partitions of X ordered by increasing value of λ such that

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

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 [7] 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 [7], 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 a subgraph X of G for w, denoted by $w(X)$, is defined as $w(X) = \sum_{u \in E(X)} w(u)$. It is then shown in [7] that the quasi-flat zones hierarchy $\mathcal{QFZ}(T, w)$ is the same as $\mathcal{QFZ}(G, w)$. It is also proved in [7] (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 an 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

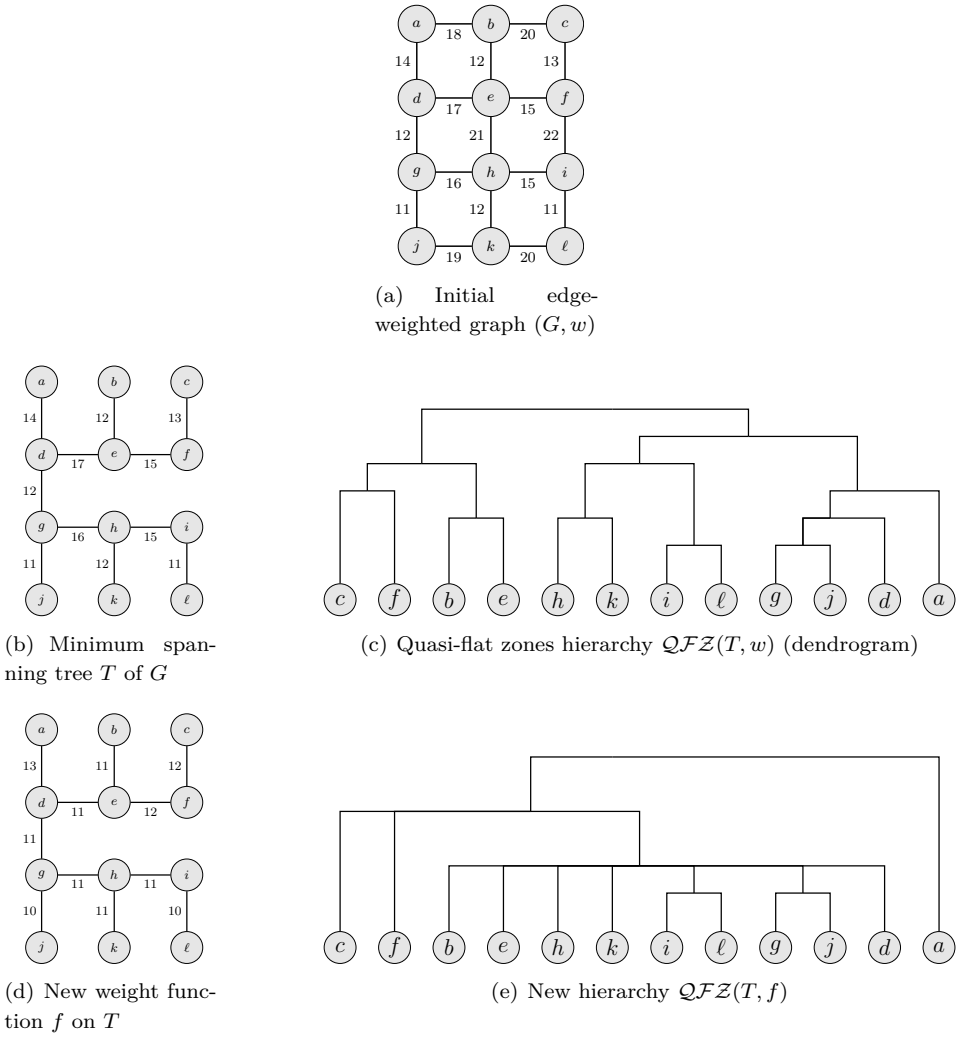


Fig. 2. Example of our graph-based hierarchical segmentation, which consists of transforming an initial hierarchy into another, by changing weights on the edges.

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 an 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 case of the method proposed by [9] later in Section 3.6. The map D associates a value 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 $D(A, B) \leq \lambda$. 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 [9], Felzenswalb *et al.* were interested in partitions of V that are neither too fine nor too coarse for a given λ, and they proposed an algorithm to extract such partitions.

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 [9] 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 [9], 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 [12]).

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 partitions 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 [7] for further details). In general, finding such a hierarchy is a complex task.

3.4 Optimal not-too-fine/not-too-coarse hierarchies

In the general introduction, we stated our interest for optimal not-too-coarse/not-too-fine hierarchies. In this section, we provide a formal definition to this problem.

If a partition $\mathbf{P}$ is a refinement of a partition $\mathbf{P}'$, we say that $\mathbf{P}$ is *smaller (or finer) than* $\mathbf{P}'$ and that $\mathbf{P}'$ is *larger (or coarser) than* $\mathbf{P}$. The set of all partitions of V, together with the relation "is larger than", is a lattice. The *infimum (resp. supremum) of two partitions* is the largest (resp. smallest) partition which is smaller (resp. larger) than the two original partitions [32, 34]. We can extend the order relation "is larger than" on partitions to the hierarchies: a hierarchy is *larger* than another if, at every level, the partition of the first hierarchy is larger than the partition of the second hierarchy. With this setting, the *infimum (resp. supremum) of two hierarchies* is given by considering, at every level, the infimum (resp. supremum) of the partitions of the two hierarchies.

Having a lattice structure for hierarchies, it is then possible to define a notion of minimal/maximal hierarchies, leading to optimization in the lattice of hierarchies. Let $\mathbb{H}$ be a set of hierarchies, we say that an element $\mathcal{H}$ of $\mathbb{H}$ is *minimal (resp. maximal) in* $\mathbb{H}$, whenever, for any hierarchy $\mathcal{G}$ of $\mathbb{H}$ such that $\mathcal{G}$ is smaller (resp. larger) than $\mathcal{H}$, we have $\mathcal{H} = \mathcal{G}$. If $\mathcal{H}$ is minimal (resp. maximal) in $\mathcal{H}$, we also say that $\mathcal{H}$ is a *smallest (resp. largest)* hierarchy among the hierarchies of $\mathbb{H}$.

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**
  $f(u) := \min\{\lambda \in \mathbb{R}^+ \mid D([\mathbf{C}(f_\lambda^V(T))]_x, [\mathbf{C}(f_\lambda^V(T))]_y) \leq \lambda\} - \varepsilon$

In this article, we are interested in the set of all hierarchies that are not-too-coarse/not-too-fine and are as large/small as possible, *i.e.*, where the regions are as large/small as possible. Therefore, our goal is to investigate the following optimization problem.

Problem 1 (Optimal not-too-coarse/not-too-fine hierarchies)**.** *Given a graph* (V, E) *and an edge-weight map* w*, consider the set* $\mathbb{H}$ *of all hierarchies which are not-too-coarse (resp. not-too-fine) and search for a hierarchy* $\mathcal{H}^\star$ *that is maximal (resp. minimal) in* $\mathbb{H}$*. Any solution to this maximization (resp. minimization) problem is called a* maximal not-too-coarse (resp. minimal not-too-fine) hierarchy (with respect to (G, w))*.*

In the next section, our goal is to propose a method which allows for finding maximal not-too-coarse hierarchies. As mentioned in Section 3.1, in this article, hierarchies are handled through weight maps. Therefore, before describing our proposed method, it is interesting to remind (see [7] for more details) that the relation "is larger/smaller than" on hierarchies can be characterized using the relation "is less/greater than" on weight maps. In particular, given two weight maps f and f' such that $f(u) \leq f'(u)$ for any u in E, the quasi-flat zone hierarchy of f (*i.e.*, $\mathcal{QFZ}(T, f)$) is larger than the quasi-flat zone hierarchy of f' (*i.e.*, $\mathcal{QFZ}(T, f')$). As it will be described in the next section, our algorithm for searching a largest not-too-coarse hierarchy consists of iteratively lowering the weights of a map (while a certain condition is satisfied) starting from a map where the weight of every edge is initialized to a maximal value, *i.e.*, a map corresponding to the smallest (not-too-coarse) hierarchy where all regions at all levels are singletons. Hence, the sequence of hierarchies, associated to this sequence of maps are ordered from small to large.

3.5 Algorithm description

The algorithm presented in this article is heuristic and does not guarantee to produce 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 [9] 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 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 sequence 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

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

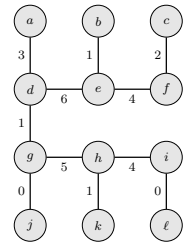


Fig. 3. Saliency map $\Phi(\mathcal{H})$ of the hierarchy $\mathcal{H}$ in Fig. 2 (c)

for all $u = \{x, y\} \in E$ where ε is a sufficiently small positive constant.

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 "merge" 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 [7] as well as [8] 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).

3.6 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 [9], 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 [9]. It is based on measuring the dissimilarity between two components (*i.e.*, regions) by comparing two types of differences: the so-called inter-component difference and within-component difference [9]. The first one is defined between two regions C_1 and C_2 by*

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

and the later one is defined for each component C by

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

Note that $Int(C) = 0$ if $C = \emptyset$. The predicate is then defined by

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

* The original version of this article, published in Mathematical Morphology - Theory and Applications 2 (2017) 55–75, unfortenately contains a mistake in the definition of $Dif(C_1, C_2)$: min was accidentally replaced by max. This typo was corrected in this version.

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 by

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

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 λ by k.

3.7 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). We simply set $\varepsilon = 1$ in this integer setting. 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.

4 Experimental setups

4.1 Compared methods, Post-processing and underlying graphs

The main aim of this and the next sections is to compare our method abbreviated by *hGB*, with its original method [9] abbreviated by *GB*. The abbreviations *hGB* and *GB* stand for "Hierarchical Graph-Based method" and "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 here is the role of such a hierarchical structure of segmentations in the procedure; indeed, we will see in the next section that the hierarchical structure imposed in *hGB* provides better results than those of *GB*. In addition, *hGB* gives the additional advantage that the hierarchical structure makes possible to apply a variety of hierarchical analyses afterwards.

The original method *GB* includes the following post-processing step: area-filtering of the segmentation results with parameter τ, which is the ratio of the component size to the image size. In order to make a fair comparison, we also apply to our method *hGB* a similar, but hierarchical, post-processing, which is using again the technique introduced in [7] and based on re-weighting a spanning tree (see Appendix A for details). In this article, the parameter value varies such that $\tau \in [0.001, 0.009]$.

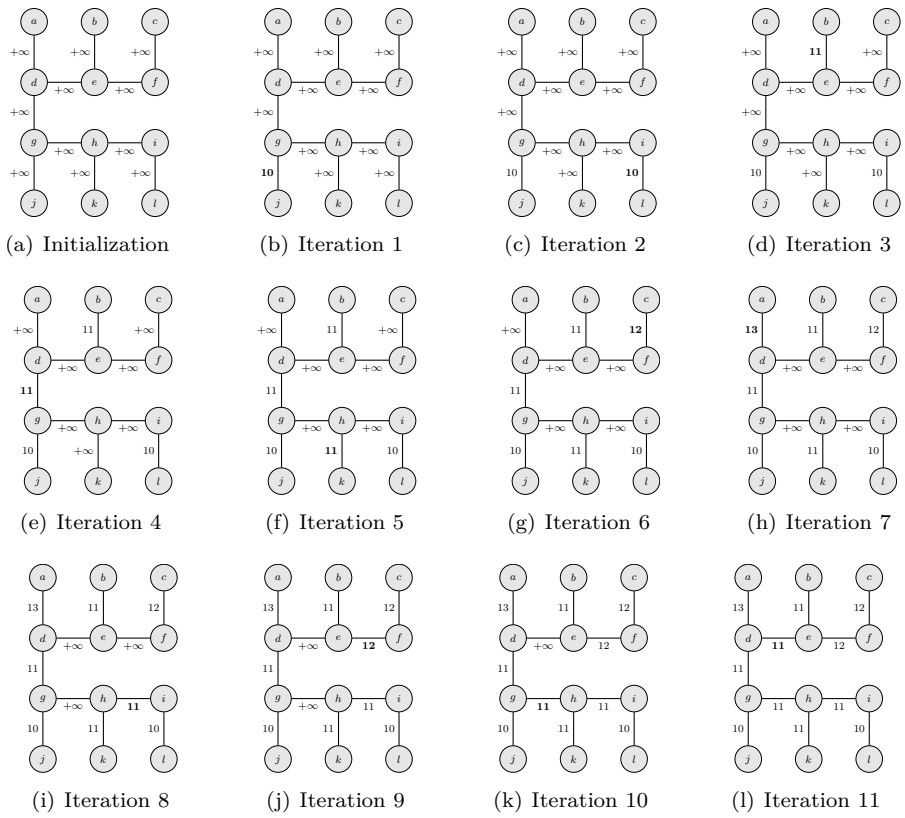


Fig. 4. Illustration of Algorithm 1 with the example in Fig. 2

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 technique to get such underlying graphs, similarly to those in [9], called *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.

4.2 Data sets and ground truths

In order to provide a comparative analysis between *GB* and *hGB*, we used the Berkeley Segmentation Dataset [21], called **BSDS500**. It 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 post-processing parameter, τ, such that those values lead to the best measure value (see Section 4.3 for the detail) on those folds. The results obtained in Section 5 are obtained on the testing fold. 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 [1, 30].

4.3 Evaluation strategies

Quantitative analysis is made for illustrating 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. 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.

4.3.1 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.

4.3.2 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 [30]. The first two precision-recall frameworks were used for evaluating hierarchical image segmentations in [1], 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 [30] then have 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.

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. It is worth to mention that, in this article, F_r is computed based on the measure of segmentation covering [1]. Note that the segmentation is perfect when $F_* = 1$ and quite different from the ground-truth when $F_* = 0$.

4.3.3 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 [1]. 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 $F^{ODS} = \max_K \operatorname{avg}_I F(I, K)$ (resp. $F^{OIS} = \operatorname{avg}_I \max_K F(I, K)$). Naturally from these definitions, F^{OIS} is generally larger than F^{ODS} as the observation scale K is better chosen for each image for OIS. In other words, the difference between OIS and ODS assesses the consistency of the hierarchy in terms of scale; if the F^{OIS} and F^{ODS} values are close, regions of equivalent perceptual importance in different images are represented at the same level of their respective hierarchies.

Concerning the area-filtering accompanied with both of the methods GB and hGB, the component size ratio with respect to the image size τ ranges from 0.001 to 0.009, as mentioned before. In order to identify the parameter value, runs were made for the best F-measure, F^{OIS} and F^{ODS}. In other words, we maximize the best F-measures for 9 values $\tau \in [0.001, 0.009]$.

5 Experimental results

5.1 Qualitative analysis

We first show Fig. 5 to illustrate some results obtained by our method when applied to some images in **BSDS500**. In order to visualize the hierarchical structures $\mathcal{H}$ of our segmentation results, we use the saliency map $\Phi(\mathcal{H})$ (see [7] for the definition, its construction from $\mathcal{H}$ and its visualization as an image) 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 easily be obtained by simple thresholding. The segmentation results of given scales are illustrated in the last row of the figure.

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

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 hGB, obtained from the original image in Fig. 1 (a) with various values of area filtering parameter. Note that we set $\tau = 0.005$ for the post-processing in both experiments.

5.2 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.

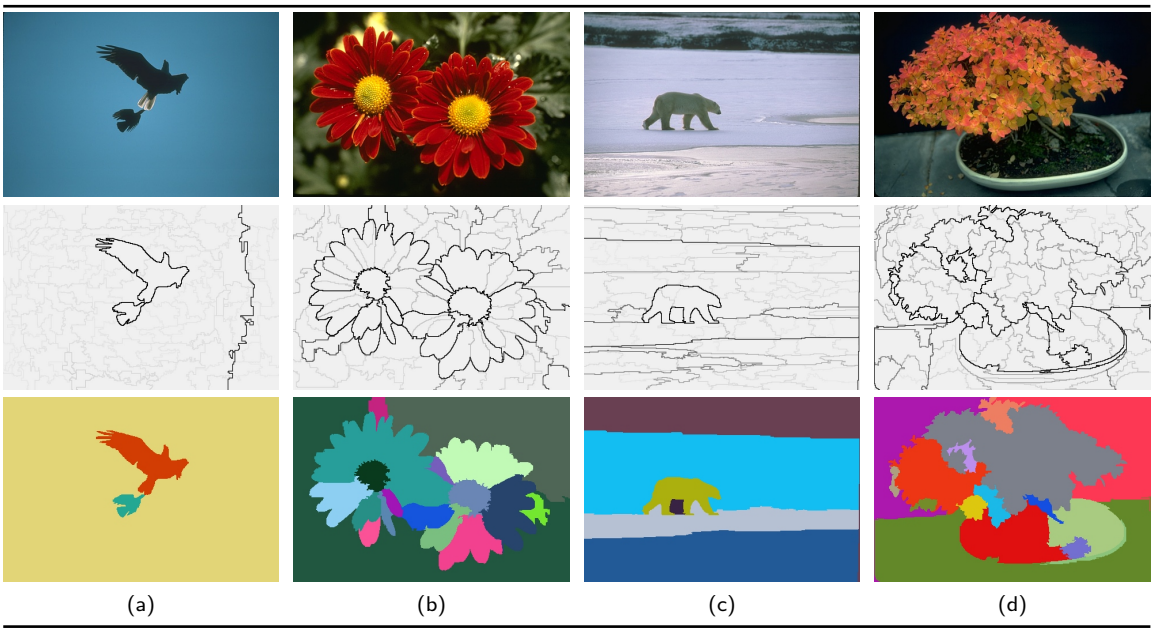


Fig. 5. Top row: some images from the Berkeley database [1]. Middle row: visualization of saliency maps of these images according to our hierarchical method. The number of different segmentations obtained from observation scales of these hierarchies are (a) 240, (b) 443, (c) 405 and (d) 429. Bottom row: examples of segmentations extracted from the hierarchies. The numbers of segmented regions are (a) 3, (b) 18, (c) 6 and (d) 16.

5.2.1 Optimal F-measure evaluations

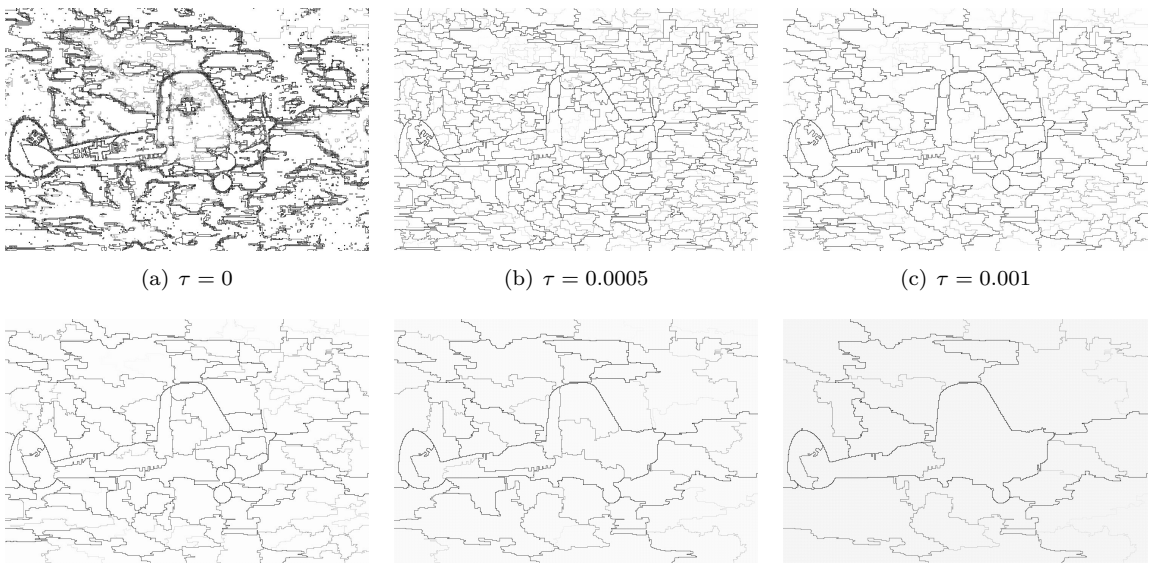
We obtained the average optimal F_r-measures for the **BSDS500** database under the OIS and ODS settings: 0.604 for hGB and 0.575 for GB under the OIS setting, and 0.492 for hGB and 0.477 for GB under the ODS setting. This first global measure tends to indicate that hGB provides better regions than GB.

To better understand the above comparison results under the OIS setting, first of all, we present the distribution of the best F_r-measures for each image, illustrated in Fig. 7 (top). Differently from the above optimal F-measures under the OIS setting, which simply presents the average of all the best F_r-measures, each of which is computed from every segmentation result, Fig. 7 (top) represents the distribution of the 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 the two distributions by using this plot. From such comparisons, we can observe that the F_r-measures of hGB are better than those of GB, as hGB provides higher median, upper quartile, lower quartile and upper whisker than GB.

Similarity to the OIS setting, we can also obtain the distribution of the F_r-measures of the best (fixed) scale K for all the images under the ODS setting, as illustrated in Fig. 7 (bottom). The comparison between the two distributions also concludes that the F_r-measures of hGB are better than GB.

5.2.2 Region quality evaluations

In addition to Fig. 7, we also made pairwise region-quality comparisons of segmentation results, as illustrated in Fig. 8, following the same presentation as in [1], in which each red point has the F-measures (with OIS (top) and ODS (bottom)), of the results of hGB and GB as its x- and y-coordinates,



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

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

Fig. 6. Visualization of saliency maps of the hierarchical segmentation results of the proposed method *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).

respectively, for each image. Here, we show the results using not only the F-measures for regions F_r but also those for objects and parts F_{op} and for boundary F_b.

The method *hGB* provides better results than *GB* mostly as there are more red dots below the line $y = x$ than those above the line except the case of using F_b-measures with OIS setting (Fig. 8 (top right)).

Figure 8 shows that *hGB* provides better results than *GB*: there are more red dots below the line $y = x$ than those above the line, except in the case of using F_b-measures with OIS setting (Fig. 8 (top right)). This observation is also confirmed by the confidence interval for each pairwise comparison with OIS setting in Fig. 8 (top): $[0.015, 0.041]$ with p-value $= 0.0000218$ for F_r (Fig. 8 (top left)), $[0.0012, 0.027]$ with p-value $= 0.0167$ for F_{op} (Fig. 8 (top middle)), and $[-0.0179, 0.00717]$ with p-value $= 0.2123324$ for F_b (Fig. 8 (top right). Similarly, the confidence interval for each pairwise comparison with ODS setting in Fig. 8 (bottom) is obtained: $[0.0085, 0.043]$ with p-value $= 0.00184$ for F_r (Fig. 8 (bottom

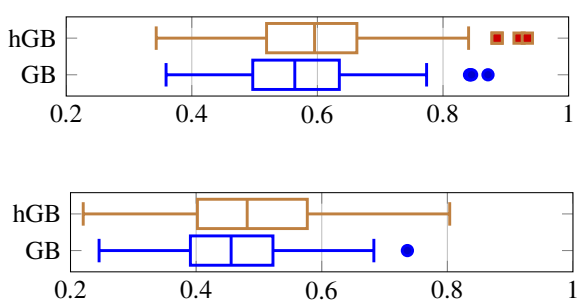


Fig. 7. Box-and-whisker plots of the distribution of the best F_r-measures (resp. the F_r-measures of the best (fixed) scale K) for all the images under the OIS (resp. ODS) setting, for the two segmentation results of *hGB* and *GB* on the database **BSDS500** (top (resp. bottom)).

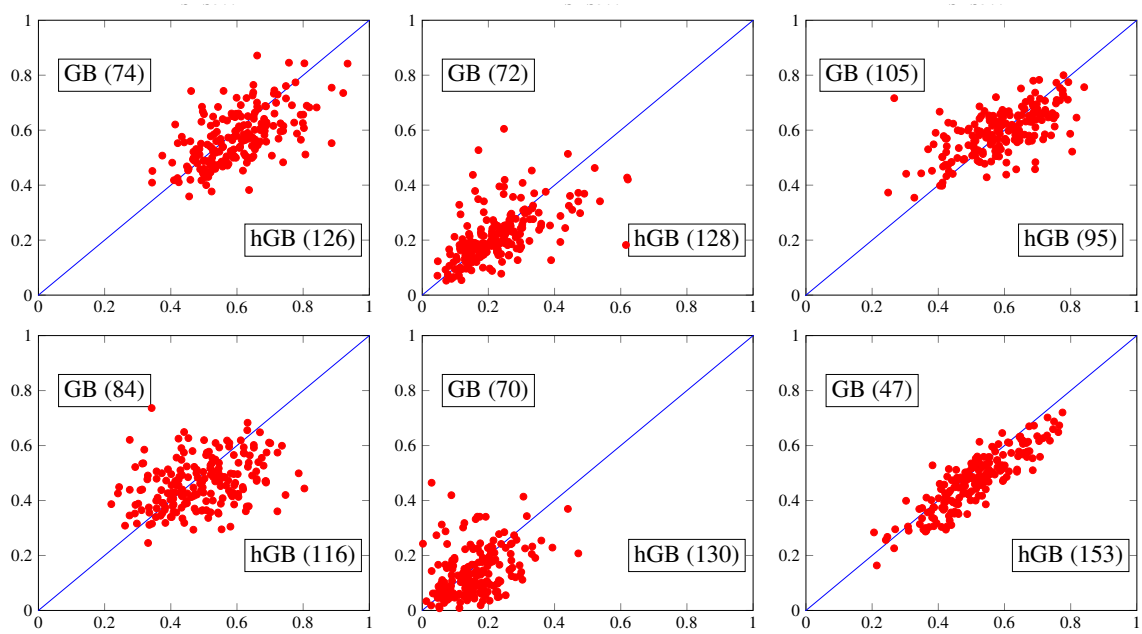


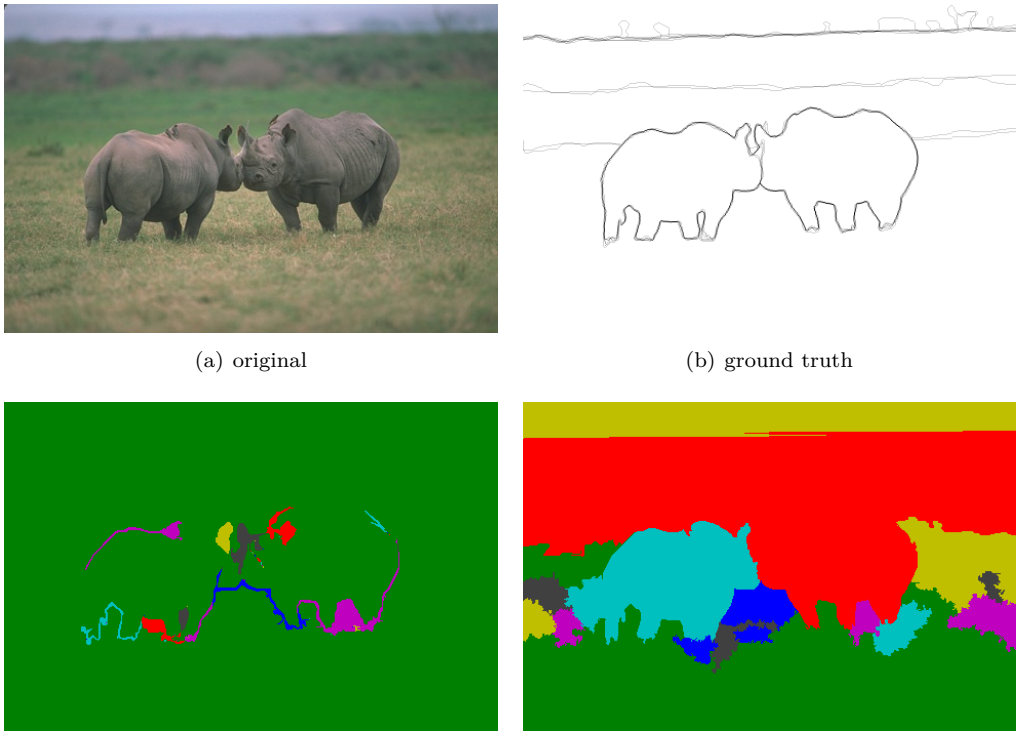
Fig. 8. Pairwise region-quality comparison of segmentation results, in which each red point has the F_r-measures (left) with OIS (resp. ODS) (top (resp. bottom)), the F_{op}-measures (middle) with OIS (resp. ODS) (top (resp. bottom)) and the F_b-measures (right) with OIS (resp. ODS) (top (resp. bottom)) of the results of *hGB* and *GB* as its x- and y-coordinates for each image. The number between parentheses for *hGB* (resp. *GB*) is the number of red points below (resp. above) the blue line.

left)), $[0.0085, 0.035]$ with p-value $= 0.000800$ for F_{op} (Fig. 8 (bottom middle)), and $[0.0319, 0.0466]$ with p-value $= 4.18 \times 10^{-21}$ for F_b (Fig. 8 (bottom right)). Note that the interpretation of the confidence interval is made as follows; if the confidence interval contains zero, then both methods are considered 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. The confidence intervals indicate that *hGB* is better than *GB* when using F_r with either OIS or ODS setting, F_{op} with either OIS or ODS setting, and F_b with ODS setting, whereas they are considered equivalent when F_b is used with OIS setting. Overall, these results lead us to conclude that *hGB* outperforms *GB*.

Lastly, we provide a discussion on the F-measures themselves with some examples of segmentation results illustrated in Figs. 9 and 10. 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 *GB* and *hGB* with respect to F_{op} are shown in (c) and (d), respectively. The results in Figs. 9 (c) and 10 (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. 9 (d) and 10 (d), while the F_b and F_r values are similar between (c) and (d) in Figs. 9 and 10 (or (c) has even higher values than (d)). A similar observation was already made in [30], and this indicates that F_{op} 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.

5.3 Running time

The efficient implementation of our method was made by using some data structures similar to the ones proposed in [12]; in particular, the management of the collection of partitions is made using Tar-



(a) original (b) ground truth

(c) result by GB: $F_r = 0.64$, $F_b = 0.62$, $F_{op} = 0.05$ (d) result by hGB: $F_r = 0.58$, $F_b = 0.57$, $F_{op} = 0.28$

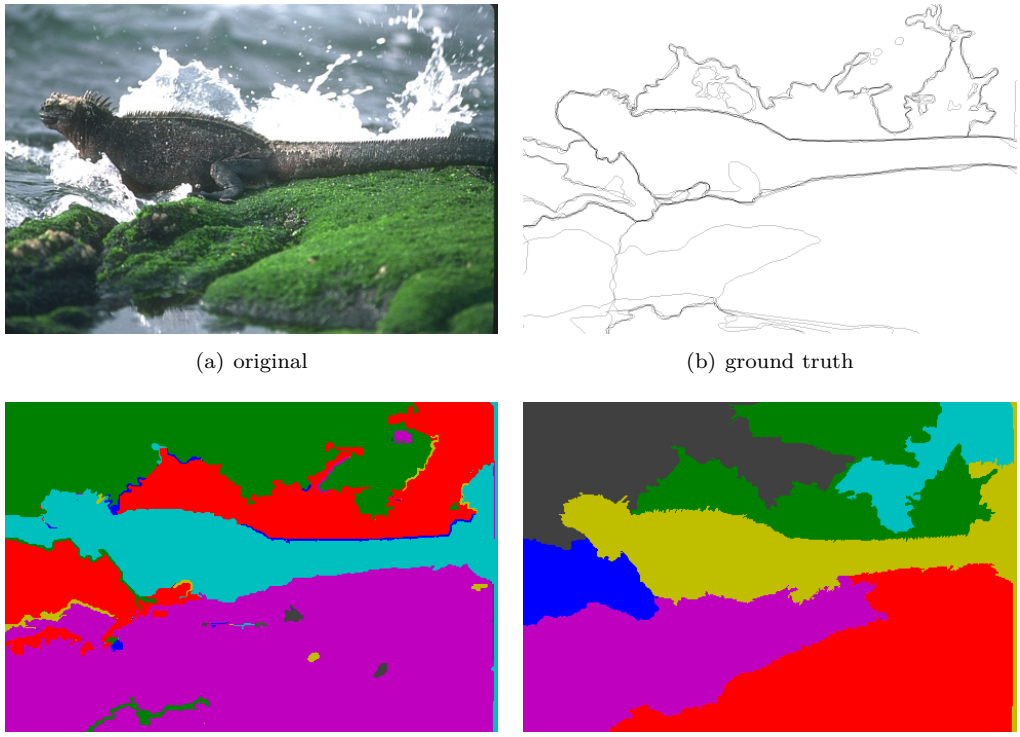
Fig. 9. Example of segmentation results of the original image (a) obtained by *GB* (c) and *hGB* (d). The F-measures, illustrated on legends, for regions (F_r), boundaries (F_b) and object and parts (F_{op}) were based on the ground truth illustrated in (b) as the saliency map.

jan's union-find algorithm [37]. Furthermore, we made some algorithmic optimizations to speed up the computations of the hierarchical scales.

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 1.37 seconds for the hierarchical segmentation result obtained by hGB and 7.84 seconds for computing 50 segmentation results with 50 different observation scales for GB, namely 0.157 seconds on average for each segmentation. Note that there are typically 500 distinct observation scales computed by the hGB method, so that it results in 0.00274 seconds on average for each segmentation obtained via hGB.

6 Conclusions

In this article, we applied the toolbox proposed in [7] 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 [9], and we performed an extensive set of experiments that demonstrates that our algorithm achieves result better than the one of [9], with the significant benefit of being much easier to tune. As far as we know, this article is a first attempt towards a general theory for hierarchizing non-hierarchical image segmentation method depending on a parameter which controls the desired level of simplification: each level of the hierarchy is "as close as possible" to the result that one would obtain with the non-hierarchical method using the corresponding scale as simplification parameter. The introduction of this hierarchization problem, as well as the proposed tools to tackle it, is an important



(a) original (b) ground truth

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

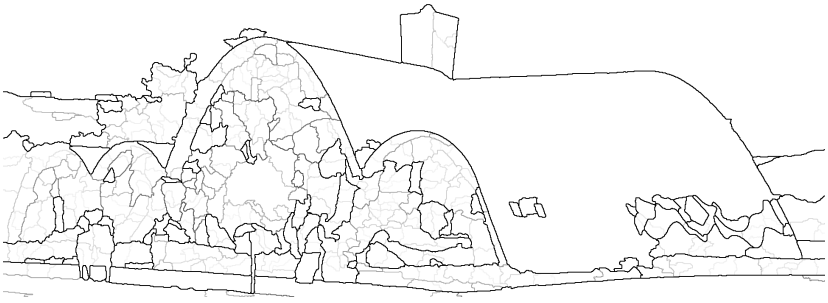
Fig. 10. Example of segmentation results of the original image (a) obtained by *GB* (c) and *hGB* (d). The F-measures, illustrated on legends, for regions (F_r), boundaries (F_b) and object and parts (F_{op}) were based on the ground truth illustrated in (b) as the saliency map.

contribution of the present article. Indeed, with the hierarchized version of a segmentation method, the user can just select the level in the hierarchy, controlling the desired number of regions or can leverage on any of the tools introduced in hierarchical analysis. An example of such tool is to consider non-horizontal cuts (*i.e.*, partitions made of regions taken at different levels of the hierarchy) rather than horizontal ones (*i.e.*, partitions made of regions which are all taken at the same level of the hierarchy). The regularization effect of such process, when the non-horizontal cut is chosen as one that minimizes the Mumford-Shah energy [10] is illustrated in Fig. 11.

We believe such results are an incentive to continue developing the generic tools proposed in [7], and investigating more deeply the optimization problem and the heuristic algorithm proposed in this article. Many other criteria, such as the one proposed in [27], can be included in our framework [13], despite being significantly more complex. Other applications, such as dealing with video, are also possible, and in some cases, our approach can provide state-of-the-art results [36]. We also envision extending such approaches for classification problems (*i.e.* non image data), as proposed in [7]. 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 [4].

For all these research directions, the question of evaluation is a fundamental one. In this article, 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. A considerable amount of work is nowadays devoted to the construction of a sound eval-

(a) original image



(b) hierarchical segmentation (saliency map)

(c) segmentation by horizontal cut

(d) segmentation by non-horizontal cut

Fig. 11. Hierarchical segmentation result and some examples of its usage: the original image (a), its hierarchical segmentation result visualized by the saliency map, obtained by *hGB* (b), a segmentation result obtained by horizontal cut (c) and non-horizontal cut [10] (d). Both (c) and (d) contains 110 regions.

uation framework [1, 29–31]. 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, Brazilian Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (program CAPES/PVE: grant 064965/2014-01, and program CAPES/COFECUB: grant 592/08), and Brazilian Conselho Nacional de Desenvolvimento Científico e Tecnológico.

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 $f'(u) := 0$;

References

[1] 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). http://doi.ieeecomputersociety.org/10.1109/TPAMI.2010.161

[2] Beucher, S.: Watershed, hierarchical segmentation and waterfall algorithm. In: J. Serra, P. Soille (eds.) Mathematical Morphology and Its Applications to Image Processing, *Computational Imaging and Vision*, vol. 2, pp. 69–76. Kluwer (1994)

[3] 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)

[4] 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)

[5] 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)

[6] Cousty, J., Najman, L., Kenmochi, Y., Guimarães, S.: New characterizations of minimum spanning trees and of saliency maps based on quasi-flat zones. In: Mathematical Morphology and Its Applications to Signal and Image Processing, pp. 205–216. Springer (2015)

[7] Cousty, J., Najman, L., Kenmochi, Y., Guimarães, S.: Hierarchical segmentations with graphs: Quasi-flat zones, minimum spanning trees, and saliency maps. Journal of Mathematical Imaging and Vision (2017). 10.1007/s10851-017-0768-7. URL https://doi.org/10.1007/s10851-017-0768-7

[8] 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)

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

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

[11] 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)

[12] 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)

[13] 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). 10.1007/978-3-319-23234-8_3. URL http://dx.doi.org/10.1007/978-3-319-23234-8_3

[14] Haxhimusa, Y., Ion, A., Kropatsch, W.G.: Irregular Pyramid Segmentations with Stochastic Graph Decimation Strategies, pp. 277–286. Springer Berlin Heidelberg (2006)

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

[16] 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). 10.1007/978-3-540-27868-9_36. URL http://dx.doi.org/10.1007/978-3-540-27868-9_36

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

[18] Laurent Najman, M.S.: Geodesic saliency of watershed contours and hierarchical segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **18**(12), 1163–1173 (1996)

[19] Marfil, R., Molina-Tanco, L., Bandera, A., Rodríguez, J., Sandoval, F.: Pyramid segmentation algorithms revisited. Pattern Recognition **39**(8), 1430 – 1451 (2006)

[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). 10.1109/TPAMI.2004.1273918. URL http://dx.doi.org/10.1109/TPAMI.2004.1273918

[22] Meyer, F.: The Dynamics of Minima and Contours, pp. 329–336. Springer US, Boston, MA (1996)

[23] Meyer, F., Maragos, P.: Morphological Scale-Space Representation with Levelings, pp. 187–198. Springer Berlin Heidelberg (1999)

[24] 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)

[25] Nacken, P.F.: Image segmentation by connectivity preserving relinking in hierarchical graph structures. Pattern Recognition **28**(6), 907 – 920 (1995)

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

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

[28] Pavildis, T.: Structural pattern recognition. Springer-Verlag (1977)

[29] 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 Inter-

national Publishing (2015)

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

[31] 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

[32] Ronse, C.: Ordering partial partitions for image segmentation and filtering: Merging, creating and inflating blocks. Journal of Mathematical Imaging and Vision **49**(1), 202–233 (2014)

[33] Salembier, P., Garrido, L.: Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. Image Processing, IEEE Transactions on **9**(4), 561–576 (2000). 10.1109/83.841934

[34] Serra, J.: A lattice approach to image segmentation. Journal of Mathematical Imaging and Vision **24**(1), 83–130 (2006)

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

[36] 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). 10.1016/j.patrec.2014.02.016. URL http://dx.doi.org/10.1016/j.patrec.2014.02.016

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

[38] 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

[39] 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)

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