A study of hierarchical watersheds on graphs with applications to image segmentation

Deise Santana Maia

UNIVERSITE PARIS-EST
ECOLE DOCTORALE MSTIC

Discipline : Informatique

Deise Santana Maia

A study of hierarchical watersheds on graphs with applications to image segmentation

Soutenue publiquement à ESIEE Paris le 10/12/2019 devant le jury composé de :

Jesus ANGULO	MINES ParisTech	Raporteur
Gunilla BORGEFORS	Uppsala University	Rapportrice
Jean COUSTY	Université Paris-Est	Co-encadrant
Mauro DALLA MURA	Institut Polytechnique de Grenoble	Examinateur
Bertrand KERAUTRET	Université Lumière Lyon 2	Examinateur et Président
Laurent NAJMAN	Université Paris-Est	Directeur de thèse
Benjamin PERRET	Université Paris-Est	Co-encadrant

Acknowledgments

I am grateful for all the professors and mentors that guided my education during my years as a bachelor, master and PhD student. Thank you Roque Trindade and Alexsandra Oliveira for inspiring me to follow an academic career after proposing my first project on image processing and mathematical morphology. Thank you Arnaldo Araujo for accepting being my master advisor and for helping me to pursue my studies as an exchange student at Université Paris-Est. Finally, my special thanks go to my PhD advisors Jean Cousty, Laurent Najman and Benjamin Perret. I feel very lucky and grateful to have collaborated with you. You have been helpful and supportive all way through my PhD studies, always ready to revise my writing, to discuss new ideas, to help me improving my speech skills... Most importantly, you helped me to be more rigorous and confident on my work.

Thank you all the members of the jury (Jesus Angulo, Gunilla Borgefors, Mauro Dalla Mura and Bertrand Keuratret) for accepting to revise my manuscript and to take part in my PhD defense.

To all my colleagues and friends who were present at some point during those years, thank you for sharing nice moments and the ups and downs of being a PhD student. Thank you Bruno Jartoux, Clara Jaquet, Daniel Antunes, Diane Genest, Duc Nguyen, Edward Cayllahua, Elias Barbudo, Evgeny Chzhen, Gisela Domej, Kacper Pluta, Karla Otiniano, Ketan Bacchuwar, Lama Tarsissi, Monika Csikos, Rosembergue Rodrigues, Stéphane Breuils, Thanh Xuan Nguyen, Tina Malalanirainy and Yukiko Kenmochi.

To my dear Thuy, thank you for your love, patience and for always being there for me. Thank you also for our interesting discussions on watersheds, which inspired me to investigate new ideas on this topic.

Last but not least, I thank my family for their unconditional love and all the efforts they made so I could be here today. I thank my father Jaime, for being an amazing, caring and loving person, who did everything that he could to give me a good education. He always hoped that everything would be better one day, and, thank to his hard work, it did get better! I thank my mother Emiliana for leading by example that there is always something else that you could learn and that it is never too late to do that. Thank you, tia Graça, for being my personal cheerleader :). And a special thanks to my sisters Daniela and Denise for being the best sisters and my best childhood friends. Thank you Denise for your positive attitude in life and for always being ready to help people around you. Thank you Daniela (*in memoriam*) for always taking care of me. I dedicate this thesis to you.

ABSTRACT

The wide literature on graph theory invites numerous problems to be modeled in the framework of graphs. In particular, clustering and segmentation algorithms designed in this framework can be applied to solve problems in various domains, including image processing, which is the main field of application investigated in this thesis. In this work, we focus on a semi-supervised segmentation tool widely studied in mathematical morphology and used in image analysis applications, namely the watershed transform. We explore the notion of a hierarchical watershed, which is a multiscale extension of the notion of watershed allowing to describe an image or, more generally, a dataset with partitions at several detail levels. The main contributions of this study are the following:

- *Recognition of hierarchical watersheds:* we propose a characterization of hierarchical watersheds which leads to an efficient algorithm to determine if a hierarchy is a hierarchical watershed of a given edge-weighted graph.

- *Watersheding operator:* we introduce the watersheding operator, which, given an edge-weighted graph, maps any hierarchy of partitions into a hierarchical watershed of this edge-weighted graph. We show that this operator is idempotent and its fixed points are the hierarchical watersheds. We also propose an efficient algorithm to compute the result of this operator.

- *Probability of hierarchical watersheds:* we propose and study a notion of probability of hierarchical watersheds, and we design an algorithm to compute the probability of a hierarchical watershed. Furthermore, we present algorithms to compute the hierarchical watersheds of maximal and minimal probabilities of a given weighted graph.

- *Combination of hierarchies:* we investigate a family of operators to combine hierarchies of partitions and study the properties of these operators when applied to hierarchical watersheds. In particular, we prove that, under certain conditions, the family of hierarchical watersheds is closed for the combination operator.

- *Evaluation of hierarchies:* we propose an evaluation framework of hierarchies, which is further used to assess hierarchical watersheds and combinations of hierarchies.

In conclusion, this thesis reviews existing and introduces new properties and algorithms related to hierarchical watersheds, showing the theoretical richness of this framework and providing insightful view for its applications in image analysis and computer vision and, more generally, for data processing and machine learning.

Résumé

La littérature abondante sur la théorie des graphes invite de nombreux problèmes à être modélisés dans ce cadre. En particulier, les algorithmes de regroupement et de segmentation conçus dans ce cadre peuvent être utilisés pour résoudre des problèmes dans de nombreux domaines tels que l'analyse d'image qui est le principal domaine d'application de cette thèse. Dans ce travail, nous nous concentrons sur un outil de segmentation semi-supervisé largement étudié dans la morphologie mathematique et appliqué à l'analyse d'image, notamment les Ligne de Partage des Eaux (LPE). Nous étudions la notion de hiérarchie de LPE, qui est une extension multi-échelle de la notion de LPE permettant de décrire une image ou, plus généralement, un ensemble de donnés par des partitions à plusieurs niveaux de détail. Les contributions principales de cette étude sont les suivantes :

- Reconnaissance de hiérarchies de LPE : nous proposons une caractérisation des hiérarchies de LPE qui mène à un algorithme efficace pour déterminer si une hiérarchie est une hiérarchie de LPE d'un graphe donné.

- Opérateur *watersheding* : nous présentons l'opérateur *watersheding*, qui, étant donné un graphe pondéré, associe n'importe quelle hiérarchie à une hiérarchie de LPE de ce graphe. Nous montrons que cet opérateur est idempotent et que ses points fixes sont les hiérarchies de LPE. Nous proposons également un algorithme efficace pour calculer le résultat de cet opérateur.

- Probabilité de hiérarchies de LPE : nous proposons et étudions une notion de probabilité d'une hiérarchie de LPE, et nous concevons un algorithme pour calculer la probabilité d'une hiérarchie de LPE. De plus, nous présentons des algorithmes pour calculer des hiérarchies de LPE de probabilité minimale et maximale pour un graphe pondéré donné.

- Combinaison de hiérarchies : nous étudions une famille d'opérateurs pour combiner des hiérarchies de partitions et nous étudions les propriétés de ces opérateurs lorsqu'ils sont appliqués à les hiérarchies de LPE. En particulier, nous prouvons

que, dans certaines conditions, la famille des hiérarchies de LPE est fermée pour l'opérateur de combinaison.

- Évaluation de hiérarchies : nous proposons un cadre d'évaluation de hiérarchies, qui est également utilisé pour évaluer les hiérarchies de LPE et les combinaisons des hiérarchies.

En conclusion, cette thèse révise des propriétés existantes et des nouvelles propriétés liées aux hiérarchies de LPE, montrant la richesse théorique de ce cadre et fournissant une vue d'ensemble des ses applications dans l'analyse d'image et dans la vision par ordinateur et, plus généralement, dans le traitement de donnés et dans l'apprentissage automatique.

Contents

List of figures

List of tables

Chapter 1

Introduction

This thesis is a study of hierarchical watersheds in the framework of weighted graphs, covering theoretical aspects of hierarchical watersheds and experiments on digital image segmentation.

Given a set V and a set E such that E is composed of pairs of elements of V, we say that the pair (V, E) is a graph, where any element of V is called a vertex and any element of E is called an edge. The first known application of graphs was on the problem of the seven bridges of Königsberg formulated by Euler in the 18th century [30]. The problem was to determine if there exists any path that passes by all seven bridges (see Figure 1.1(a)) exactly once in such a way that the river can only be crossed through one of the bridges. Euler solved this problem using a graph representation of those bridges: each land area is represented by a vertex and each bridge is represented by an edge linking a pair of land areas, as shown in Figure 1.1(b). He proved that there is no solution to this problem. Furthermore, Euler demonstrated that the existence of a solution to other similar problems depends only on the topology of the underlying graphs rather than on the absolute geometrical positions of the bridges.

Since then, numerous optimization problems have been formulated in the framework of graphs. For instance, let us consider shortest path optimization problems. Let (V, E) be a graph and let x and y be two vertices in V. A path from x to y (in (V, E)) is a sequence $(x_1, \ldots, x_\ell)$ such that $x_1 = x$, $x_\ell = y$ and such that, for i in $\{1, \ldots, \ell - 1\}$, the edge $\{x_i, x_{i+1}\}$ is an edge in E. If there is a path from x to y in (V, E), we say that x and y are connected for (V, E). A shortest path from x to y is a path $(x_1, \ldots, x_\ell)$ such that ℓ is minimal among all paths from x to y. In practice, a vertex can represent a "real object" as a city or an antenna, or an "abstract object" such as a word in a given language or profiles of a social network. Hence, edges can link neighbouring cities, closest pairs of antennas, similar words or virtual friends/followers. In this context, solving the

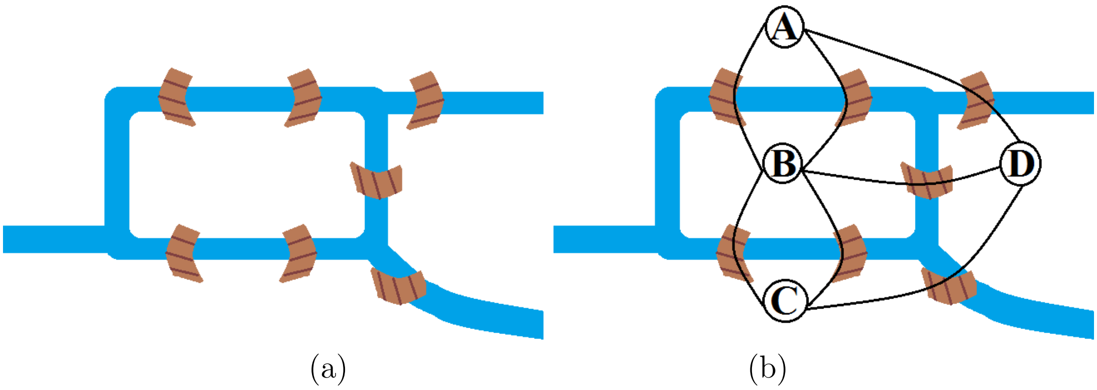


Figure 1.1: (a): The configuration of the seven bridges of Königsberg in the 18th century. (b): The representation of the seven bridges of Königsberg as a graph.

shortest path problem can help with finding the most direct path between any two cities or antennas, the least number of modifications needed to transform one word into another, or to infer "how close are two users of a social network?".

More complex problems can be formulated by assigning weights to the edges of a graph. Let $G = (V, E)$ be a graph and let w be a map from E into the set of positive real numbers $\mathbb{R}^+$. For any edge u in E, we call $w(u)$ the weight of u. We call (G, w) a edge-weighted graph. In this context, the shortest path problem can be formulated so as to take into consideration the weights of the edges in E. Let f be any function from any path $\pi = (x_1, \ldots, x_\ell)$ into the set $\mathbb{R}^+$ of positive real numbers, *e.g.* $f(\pi) = max\{w(\{x_i, x_{i+1}\}) \mid i \in \{1, \ldots, \ell - 1\}\}$ or $f(\pi) = \sum_{i=1}^{\ell} w(\{x_i, x_{i+1}\})$. Given any two vertices x and y in V, a shortest path from x to y is a path $\pi = (x_1, \ldots, x_\ell)$ from x to y such that $f(\pi)$ is minimal among all paths from x to y. Considering the examples given in the previous paragraph, the weight of an edge can represent, for example, a road length, the time to transfer a given amount of data between two antennas, the level of similarity between two words or the number of interactions between two users of a social network. Hence, incorporating weights to the edges of a graph aids to approximate the graph representation to the real world problems.

As the problems aforementioned, the large literature on graph theory invites many classification problems to be formulated in the framework of graphs. We denote by class or cluster a group of (data) points which are similar with respect to a given criterion. Let P be a set. A classification of the elements of P is the assignment of each element of P to a class (or to several classes). In the context of graphs, the set P is formalized as the set of vertices of a graph, whose edges link pair of vertices that are potentially in the same class. Let (G, w) be a edge-weighted graph. A classification of the vertices of G

is often obtained by minimizing energy functions related to the weight of edges linking vertices in the same class (or edges linking vertices in different classes). For instance, let us consider that the vertices of G represent the researchers in a given conference. We may be interested in classifying this group of researchers according to their specific area of interest. To do so, we could link each two researchers x and y by an edge whose weight is the number of articles in which x cites y plus the number of articles in which y cites x. We can infer that, in most cases, the researchers working on the same area are linked by edges of larger weights than the edges linking researchers in different areas. Hence, a classification of the vertices of G could be the result of maximizing (resp. minimizing) the sum of the edges linking vertices in a same class (resp. different classes).

We can consider two variants of the previous classification problem: (1) supervised classification: the areas of research are defined beforehand and samples of each class are used by the classification algorithm; and (2) unsupervised classification: the areas of research are deduced from the result of the classification algorithm.

Similarly to classification problems, segmentation problems are also commonly formulated in the framework of graphs. Let P be a set. A segmentation of P is a partition of P into disjoint subsets $R_1, \ldots, R_\ell$ such that every element of P is in exactly one subset R_i. Each subset of a segmentation of P is called a region of this segmentation. In practice, the set P represents a set of real or abstract objects whose similarity can be measured. Then, the elements of P are segmented by their degree of similarity. We can observe that classification and segmentation problems are related: a classification of the elements of P into disjoint classes induces a partition of P and, on the other hand, segmenting P can be a pre-processing step to classification algorithms, which is often the case. As classification problems, we can also consider two variations of segmentation methods: (1) marker-based segmentation: given a set S of disjoint subsets (markers) of P, each region of the final segmentation of P includes exactly one element of S; and (2) no markers are provided and the final regions depend on the algorithm and possibly on other parameters such as number and size of regions.

A typical segmentation problem is the segmentation of digital images. A (digital) image is a matrix of picture elements or *pixels* such that each pixel comprises the color or gray-scale information of a point in the image. Let I be an image. In the framework of edge-weighted graphs, the image I can be represented as an edge-weighted graph (G, w) such that the vertices of G correspond to the pixels of I, the edges of G link neighbouring pixels of I, and the edge weights are a measure of dissimilarity between pixels. Alternatively, the vertices of G can also represent disjoint subsets of connected pixels of I, with the resulting graph known as a Region Adjacency Graph (RAG).

Several notions related of graphs induce good solutions for practical image segmentation problems. Hence, we can profit of the efficient algorithms that have already been developed in graph theory in order to solve new application problems. To give an example, let us consider the minimum spanning forest problem. Let $((V, E), w)$ be a graph and let S be a set of disjoint subsets of the vertex set of G. A minimum spanning forest of $((V, E), w)$ rooted in S is a graph $((V, E'), w)$ such that:

1. any two vertices in a same element of S are connected for (V, E');

2. any two vertices in distinct elements of S are not connected for (V, E'); and

3. the sum of the weight of all edges in E' is minimal for all graphs for which statements 1 and 2 hold true.

In the context of image segmentation, we can see that the minimum spanning forest of $((V, E), w)$ rooted in S induces a marker-based segmentation in which the regions are determined by the vertices (pixels) that are connected in (V, E'). As we will see later, the notion of minimum spanning forests induce efficient segmentation algorithms which satisfy relevant mathematical properties. Moreover, minimum spanning forests are closely related to the segmentation method explored in this thesis: the watershed transform.

In the late 70's, the watershed transform was proposed as a powerful tool in the segmentation of gray-scale digital images. Since then, numerous definitions and algorithms to implement the watershed transform have been designed. The idea behind the watershed transform is that an image (or a weighted graph) can be visualized as a topographic surface. In this context, a set of connected pixels surrounded by pixels of strictly greater gray values (or a set of adjacent vertices/edges surrounded by vertices/edges of strictly greater weights) is a regional minimum of the surface. Each regional minimum can be associated to a zone of influence, known as a catchment basin. From any point x (pixel or vertex) in the zone of influence of a regional minimum, there is a descending path from x to this regional minimum, where a descending path is either defined as a sequence of connected pixels of non-increasing gray levels, or a sequence of vertices connected by edges of non-increasing weights.

By iteratively merging the regions of a segmentation, we produce a hierarchy of segmentations, which is a sequence of nested segmentations of an image. Hence, a hierarchy provide segmentations of an image with different levels of detail, where the segmentation in the lowest level contains the largest number of regions. When the initial segmentation is a watershed segmentation of a graph and when the merging steps are guided by a sequence of minima of this graph, we obtain a hierarchical watershed.

When formalized in the framework of weighted graphs, hierarchical watersheds are deeply linked to the optimization problem of minimum spanning trees. As a result of this link, each segmentation of a hierarchical watershed is optimal in the sense of minimum spanning forests, providing us good quality hierarchy of segmentations that optimizes a well defined objective function. Moreover, minimum spanning tree algorithms can be adapted to the computation of hierarchical watersheds, which leads to efficient algorithms to compute the latter.

Hierarchical watersheds are part of a broader family of hierarchical image representations, whose main applications include image simplification and filtering, implementation of morphological connected operators, and provision of a larger search space for object detection tasks (when compared to a single segmentation).

In this work, we study hierarchical watersheds in the framework of graphs. For visualization and evaluation purposes, we recur to the problem of digital image segmentation. However, the theoretical results introduced in this manuscript hold for arbitrary graphs and, hence, can be applied to a broader range of problems.

The remainder of this manuscript is organized as follows:

- Chapter 2 introduces the background theory of this research. We present a brief survey and the formal definitions related to each of those topics: hierarchical image representations, weighted graphs, connected hierarchies, saliency maps, morphological hierarchies (quasi-flat zones hierarchies, binary partition trees and hierarchical watersheds), and attribute based hierarchies.

- Chapter 3 proposes a characterization of hierarchical watersheds and an efficient algorithm to recognize hierarchical watersheds. Using the notions of saliency map and binary partition hierarchy by altitude ordering (a special case of binary partition trees), we present a necessary and sufficient condition for any connected hierarchy to be a hierarchical watershed.

- Chapter 4 presents the watersheding operator, which converts any hierarchy into a hierarchical watershed of a given weighted graph. This operator is idempotent and its set of fixed points is precisely the set of hierarchical watersheds. Hence, we establish the link between the watersheding operator and the problem of recognizing of hierarchical watersheds studied in Chapter 3. We also present an efficient algorithm that implements the watersheding operator and experimental results on images.

- Chapter 5 studies the probability of hierarchical watersheds. By definition, a hierarchical watershed can be computed from a sequence of minima of a weighted graph. In this chapter, we demonstrate that a hierarchical watershed can be obtained from numerous sequences of minima of a graph. We show that the number of sequences of minima associated to different hierarchical watersheds of a weighted graph may differ. In this context, we define the probability of a hierarchical watershed with respect to the number of sequences of minima that could be used to compute this hierarchy. Then, we present an efficient method to obtain the probability of a hierarchical watershed, and a characterization of the most and least probable hierarchical watersheds of a weighted graph.

- Chapter 6 introduces an evaluation framework of hierarchies of segmentations. We present three evaluation measures that summarize several aspects of a hierarchy of segmentations, including the tendency to over and under-segmentation, and the easiness of extracting objects of interest with the help of markers. This evaluation framework allows us to identify a hierarchical watershed based on a novel extinction value that outperform the classical area, dynamics and volume based hierarchical watersheds. Then, this evaluation framework is used to compare hierarchical watersheds with other morphological hierarchies.

- Chapter 7 presents theoretical and experimental results of combinations of hierarchical watersheds. We first perform a visual inspection of combinations of hierarchies. Then, using the evaluation framework introduced in Chapter 6, we show that combinations of hierarchical watershed through their saliency maps can outperform the input hierarchies. We also study properties of combinations by providing a sufficient condition for a combination to always output a flattened (simplified) hierarchical watershed. Then, we present experimental results with the algorithm to recognize hierarchical watersheds (Chapter 3) applied to combinations of hierarchical watersheds. Finally, we show the interest of applying the watersheding operator (Chapter 4) to combinations of hierarchical watersheds: evaluation scores at least as good as the combinations with the advantage of preserving the mathematical properties of hierarchical watersheds.

The results presented in this manuscript have been partially published in the following articles:

Conference proceedings:

- D. S. Maia, A. de Albuquerque Araujo, J. Cousty, L. Najman, B. Perret, and H. Talbot. Evaluation of combinations of watershed hierarchies. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 133–145. Springer, 2017.
- D. S. Maia, J. Cousty, L. Najman, and B. Perret. Recognizing hierarchical watersheds. In *International Conference on Discrete Geometry for Computer Imagery*, pages 300–313. Springer, 2019.
- D. S. Maia, J. Cousty, L. Najman, and B. Perret. Watersheding hierarchies. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 124–136. Springer, 2019.
- D. S. Maia, J. Cousty, L. Najman, and B. Perret. On the probabilities of hierarchical watersheds. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 137–149. Springer, 2019.

Journals:

- B. Perret, J. Cousty, S. J. F. Guimaraes, and D. S. Maia. Evaluation of hierarchical watersheds. *IEEE Transactions on Image Processing*, 27(4):1676–1688, 2017.
- D. S. Maia, J. Cousty, L. Najman, and B. Perret. Properties of combinations of hierarchical watersheds. Under review. 2019.
- D. S. Maia, J. Cousty, L. Najman, and B. Perret. Characterization of graph based hierarchical watersheds: theory and algorithm. Under review. 2019.

Chapter 2

Hierarchies and Graphs

In this chapter, we present the background theory that led to the development of this thesis. We review graphs and hierarchies of partitions, in particular the family of hierarchies used in this research: quasi-flat zones hierarchies, binary partition trees, hierarchical watersheds and attribute based hierarchies.

Remark. *The notations presented in this chapter are used all along the manuscript.*

2.1 Graphs

A *graph* is a pair $G = (V, E)$, where V is a finite set and E is a set of pairs of distinct elements of V, *i.e.*, $E \subseteq \{\{x, y\} \subseteq V \mid x \neq y\}$. Each element of V is called a *vertex (of G)*, and each element of E is called an *edge (of G)*. To simplify the notations, the set of vertices and edges of a graph G will be also denoted by $V(G)$ and $E(G)$, respectively.

Let $G = (V, E)$ be a graph and let X be a subset of V. A sequence $\pi = (x_0, \ldots, x_n)$ of elements of X is a *path (in X) from x_0 to x_n* if $\{x_{i-1}, x_i\}$ is an edge of G for any i in $\{1, \ldots, n\}$. Given a path $\pi = (x_0, \ldots, x_n)$ from a vertex x_0 to a vertex x_n in V, for any edge $u = \{x_{i-1}, x_i\}$ for any i in $\{1, \ldots, n\}$, we say that *u is an edge in π*. The subset X of V is said to be *connected (for G)* if, for any x and y in X, there exists a path from x to y. The subset X is a *connected component* of G if X is connected and if, for any connected subset Y of V, if $X \subseteq Y$, then we have $X = Y$. In the following, we denote by $CC(G)$ the set of all connected components of G.

Let G be a graph. If w is a map from the edge set of G to the set $\mathbb{R}$ of real numbers, then the pair (G, w) is called an *(edge) weighted graph* (see Figure 2.1(a)). If (G, w) is a weighted graph, for any edge u of G, the value $w(u)$ is called the *weight of u (for w)*.

We say that the graph $G = (V, E)$ *is a forest* if, for any edge u in E, the number of

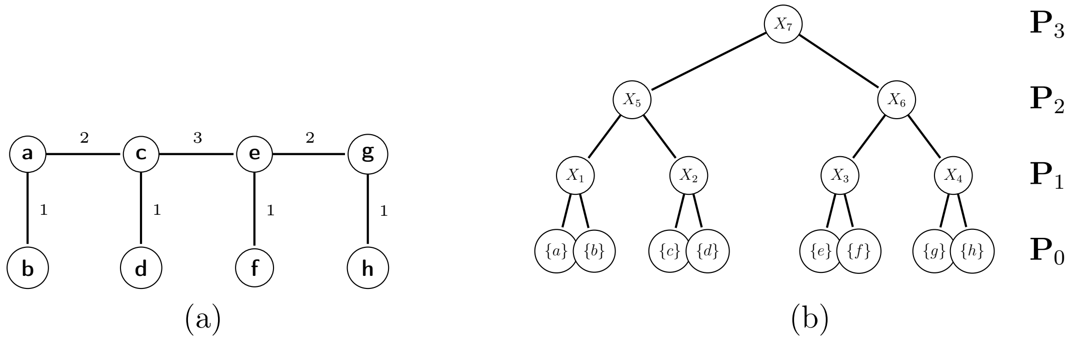


Figure 2.1: (a): A weighted graph (G, w). (b): A representation of a hierarchy of partitions $\mathcal{H} = (\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3)$ on the set $\{a, b, c, d, e, f, g, h\}$.

connected components of the graph $(V, E \setminus \{u\})$ is greater than the number of connected components of G. Given another graph G', we say that G' *is a subgraph of* G, denoted by $G' \sqsubseteq G$, if $V(G')$ is a subset of V and $E(G')$ is a subset of E. Let G'' be a subgraph of G and let G' be a subgraph of G''. The graph G'' is a *Minimum Spanning Forest (MSF) of* G *rooted in* G' if:

1. the graphs G and G'' have the same set of vertices, *i.e.*, $V(G'') = V$; and
2. each connected component of G'' includes exactly one connected component of G'; and
3. the sum of the weight of the edges of G'' is minimal among all subgraphs of G for which the above conditions 1 and 2 hold true.

A MSF of (G, w) rooted in a single vertex of G is a tree (connected forest) called a *Minimum Spanning Tree (MST) of* (G, w).

Let (G, w) be a weighted graph and let k be a value in $\mathbb{R}$. A connected subgraph G' of G is a *(regional) minimum* (of w) at level k if:

1. the set of edges $E(G')$ of G' is not empty; and
2. for any edge u in $E(G')$, the weight of u is equal to k; and
3. for any edge $\{x, y\}$ in $E \setminus E(G')$ such that $|\{x, y\} \cap V(G')| \geq 1$, the weight of $\{x, y\}$ is strictly greater than k.

2.2 Hierarchies of partitions

In this section, we first introduce notations and definitions related to hierarchies of partitions. Then, we review partitions and hierarchies of partitions in the context of digital image processing and analysis.

2.2.1 Notations and definitions

Let V be a set. A *partition* (of V) is 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* $\mathbf{P}$. Let $\mathbf{P}_1$ and $\mathbf{P}_2$ be two partitions. We say that $\mathbf{P}_1$ is a *refinement* of $\mathbf{P}_2$ if every element of $\mathbf{P}_1$ is included in an element of $\mathbf{P}_2$. A *hierarchy (of partitions)* is a sequence $\mathcal{H} = (\mathbf{P}_0, \ldots, \mathbf{P}_\ell)$ of partitions such that $\mathbf{P}_{i-1}$ is a refinement of $\mathbf{P}_i$, for any i in $\{1, \ldots, \ell\}$ and such that $\mathbf{P}_n = \{V\}$. Let $\mathcal{H} = (\mathbf{P}_0, \ldots, \mathbf{P}_\ell)$ be a hierarchy of partitions. Any region of a partition $\mathbf{P}$ of $\mathcal{H}$ is called a *region of* $\mathcal{H}$. The set of all regions of $\mathcal{H}$ is denoted by $\mathscr{R}(\mathcal{H})$.

A hierarchy of partitions can be represented as a tree whose nodes correspond to regions, as shown in Figure 2.1(b). Given a hierarchy $\mathcal{H}$ and two regions X and Y of $\mathcal{H}$, we say that X *is a parent of* Y *(or that* Y *is a child of* X*)* if $Y \subset X$ and X is minimal for this property, *i.e.*, if there is a region Z such that $Y \subseteq Z \subset X$, then we have $Y = Z$. It can be seen that any region $X \neq V$ of $\mathcal{H}$ has exactly one parent. For any region X such that $X \neq V$, we write $parent(X) = Y$ where Y is the unique parent of X. For any region R of $\mathcal{H}$, if R is not the parent of any region of $\mathcal{H}$, we say that R *is a leaf region (of* $\mathcal{H}$*)*. Otherwise, we say that R *is a non-leaf region (of* $\mathcal{H}$*)*.

We illustrate a hierarchy of partitions $\mathcal{H}$ in Figure 2.1(b). The regions of the hierarchy $\mathcal{H}$ are represented by nodes on a tree. Each region of $\mathcal{H}$ is linked to its parents (and to its children) by straight lines.

Let $G = (V, E)$ be a graph. *A partition of* V *is connected for* G if each of its regions is connected and *a hierarchy on* V *is connected (for* G*)* if every one of its partitions is connected. For example, the hierarchy of Figure 2.1(b) is connected for the graph of Figure 2.1(a).

2.2.2 Partitions in the context of digital images

A digital image is a numeric representation of an image: a matrix or set of picture elements or *pixels*, where each pixel carries the colorimetric information of a point in the image (see Figure 2.2). The information associated to each pixel varies depending on the nature of the image representation. In gray-scale images, a pixel can be associated to a single value that indicates the gray-level of this pixel - brighter pixels being assigned to greater values. In turn, a pixel of a color image can be represented as a combination of the levels of red (R), blue (B) and green (G) colors in this pixel. The latter representation corresponds to a vector in the RGB color space.

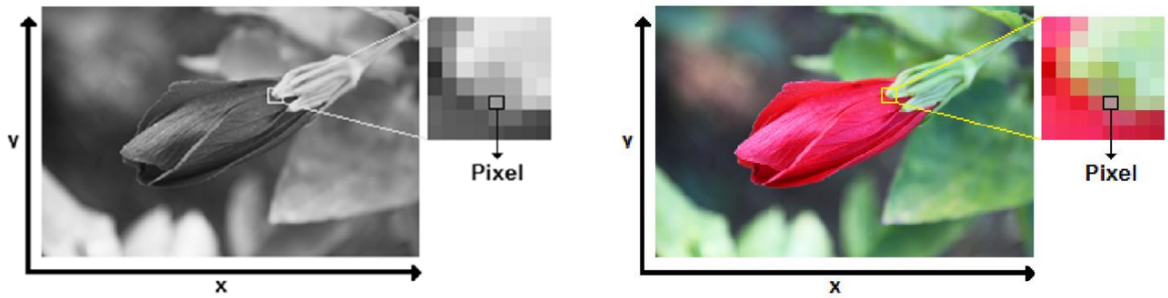


Figure 2.2: A gray-scale and a color image.

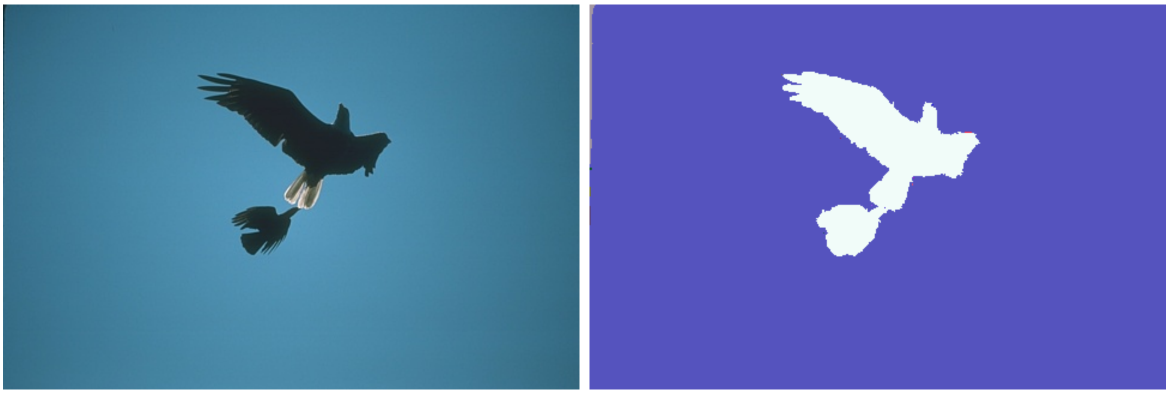


Figure 2.3: An image I and a partition of I into two regions.

Let I be an image and let $\{p_0, \ldots, p_n\}$ be the set of pixels of I. A partial partition of I is a set of disjoint subsets of $\{p_0, \ldots, p_n\}$. A (total) partition or segmentation of I is a set of disjoint subsets of $\{p_0, \ldots, p_n\}$ such that every pixel of I belongs to an element of this partition. Each element of an image partition is called a region of this partition. Along this manuscript, the terms partition and segmentation will be used interchangeably. An image segmentation is illustrated in Figure 2.3.

The need for image segmentation arose with the various applications of digital images in research fields such as biology, medicine and astronomy. Image segmentation is usually a pre-processing step to other image processing and analysis tasks, including image filtering and simplification, object detection, object tracking and scene labeling. As the size of images to be processed increases, manual segmentation becomes an onerous task.

In the early days of image segmentation, heuristic techniques for image segmentation have been employed. For instance, we can cite low-level pixels classification techniques such as thresholding and histogram analysis, in which pixels are segmented based solely on their colorimetric information.

In the pioneer work of Zahn [107], the author proposed a clustering method to segment arbitrary sets of points based on the distance between those points. When applied to image segmentation, this technique takes into consideration not only the local information of a pixel but also its position with respect to other similar pixels. Following a similar idea, the authors of [35] propose a greedy graph-based segmentation method that relies on the dissimilarity between pixels of a region and on the dissimilarity between neighbouring regions of a segmentation. The latter method uses a greedy approach to optimize a global energy, which is also the case of the segmentation technique investigated in this research: the watershed transform.

The watershed segmentation was first studied in [12] and, since then, numerous definitions have been proposed [2, 22]. The idea underlying this technique is that a node (vertex) or edge weighted graph can be visualized as a topographic surface in which the node and edge weights determine the altitude of the points on the surface. In geography, a catchment basin is a region whose collected water drain to a common point (*e.g.* a sea), and the watersheds are the dividing lines between neighbour catchment basins. Each catchment basin is associated to a regional minimum, which is a plateau surrounded by points of greater altitude. A point in the surface belongs to a given catchment basin if there exists a descending path from this point to the regional minimum in this catchment basins. In the context of node and edge weighted graphs, a regional minimum is a subgraph (or a subset of vertices) of uniform weight and surrounded by nodes or edges of greater weights. The watershed transform segments the vertices of a weighted graph into its catchment basins. This idea can be applied to the segmentation of gray-scale images by either computing an image gradient represented as a weighted graph or by considering that the altitudes of the topographic surface are given by the pixel gray-levels. More details on the watershed segmentation are given later in Section 2.6.

Image segmentation is an ill-posed problem as it does not have a fixed optimal solution for all applications. Still, efforts have been made to establish ground-truths for large image datasets, which can be further used as a reference to image segmentation algorithms. Such datasets include the Berkeley Segmentation Dataset and Benchmark (BSDS500) [63], Grabcut [16], Weizman [1], Pacal Context [71] and COCO [54].

2.2.3 Hierarchies of partitions in the context of digital images

As stated in the previous section, there is no global optimal segmentation of an image. For different tasks, segmentations with distinct levels of detail may be required. In this context, hierarchies of image segmentations arise as an all-purpose tool for image segmentation. More generally, hierarchies of image segmentations are part of a broader group of hierarchical image representations, which also include hierarchies of partial partitions and inclusion hierarchies. As discussed later in this section, the use of hierarchical image representations goes beyond image segmentation.

In the remainder of this section, we review well-known hierarchical image representations and their applications.

Since the early work of [78] on a splitting and merging hierarchical partition algorithm, several methods to compute and process hierarchies of partitions have been proposed. Among the image processing tasks aided by hierarchies of partitions, we cite image simplification, filtering, and segmentation.

In [13], the author propose an algorithm, called waterfall algorithm, to overcome the oversegmentation resulting from watershed segmentations. In his algorithm, the initial regions of a watershed segmentation are iteratively merged until a simplified segmentation is obtained. The intermediate segmentations produced by this method compose a hierarchy of partitions.

In [48], the authors proposed a hierarchical segmentation method based on the graph-based segmentation introduced in [35]. The algorithm proposed in [35] receives as input a parameter to control the size of regions and the dissimilarity between the regions of the resulting segmentation. As the causality and location properties do not hold for the segmentations produced by [35] using increasing parameters, the authors of [48] propose an adaptation of this method.

In the family of morphological hierarchies with large applications to image filtering and simplification, we can cite min-trees, max-trees, tree of shapes, quasi-flat zones hierarchies, binary partition trees and hierarchical watersheds [72, 13, 87, 65, 24, 27, 74], which will be explored next.

Let I be a gray-scale image. A level-set of I is a subset of the pixels of I with gray values greater than a given threshold parameter λ. Any gray-scale image can be equally represented by its level-sets. The min-tree and max-tree are dual representations of the level-sets of an image: the max-tree of an image I is composed of the connected components of the level-sets of I while that the min-tree of I is composed of the connected components of the complement of the level-sets of I. Therefore, the leaf regions of a max-tree (resp. min-tree) are the regional maxima (resp. minima) of an image. Those trees are widely used in the implementation of connected operators [88, 89], the max-tree (resp. min-tree) being useful to compute anti-extensive (resp. extensive) operators.

Let I be a gray-scale image. A flat zone of I is a maximal connected set of pixels of I with uniform gray values. Connected operators act by removing flat-zones and, therefore, do not create any new contours in the image. A quasi-flat zone is a connected set of pixels whose gray-level difference of neighbouring pixels is limited by a given value threshold λ.

A quasi-flat zone hierarchy is a sequence of segmentations composed of flat-zones of an image with increasing values for the parameter λ. In the context of edge-weighted graphs, a quasi-flat zone is a connected set of vertices such that the difference between the weight of any two adjacent edges is limited to a given value λ. The connection between quasi-flat zone hierarchies and other morphological hierarchies has been studied in [27]. Moreover, quasi-flat zones hierarchies are linked to a dual representation of hierarchies of partitions known as saliency maps, which is explained later in Section 2.4.

The tree of shapes [65] is a hierarchical image representation based on the inclusion relationship between the connected components of the level-sets (and of the complement of the level-sets) of an image. They are self-dual and contrast-invariant. Furthermore, the tree of shapes is a compact representation of the max-tree and min-tree of an image since any region of those two hierarchical representations can be found in the tree of shapes. Properties and applications of the tree of shapes have been studied in [104] and an efficient algorithm to obtain a tree of shapes is given in [39].

Binary partitions trees were first proposed by Salembier and Garrido [87] as a tool for simplifying, segmenting and extracting information from images. The construction of this hierarchy relies on the notions of merging order, merging criterion and region model. Given any segmentation $\mathbf{P}$, a binary partition hierarchy is constructed by merging the regions of $\mathbf{P}$ following a given merging order defined on the regions of $\mathbf{P}$. Each region built along this process is represented according to a region model as, for example, the average gray-level of the pixels belonging to a region. The merging criterion, *e.g.* the color homogeneity between two regions, determines if any two neighbouring regions should be merged. Particular cases of this hierarchy have been studied under several names, such as α-tree [77] and binary partition hierarchy by altitude ordering [27]. An extension of binary partition hierarchies applied to multiple images and multiple criteria was proposed in [86].

Outside the group of morphological hierarchies, several high-quality hierarchical segmentation methods have been proposed [5, 7, 60].

In [5], the authors introduce a multiscale contour detector that combines multiple contour cues: brightness, color and texture gradient. Then, the output of their contour detector is used to obtain a segmentation through their method called oriented watershed transform. The oriented watershed transform outputs weighted boundaries which are further used to compute a hierarchy of partitions.

In [7], the authors propose a hierarchical segmentation method based on the combination of hierarchies computed from different resolutions of the same image. For each resolution, they compute the normalized cuts of the image contours, which are based on the same contour cues used by [5]. Then, those normalized cuts are combined into a single Ultrametric Contour Map (UCM), which is a dual representation of a hierarchy of partitions and also known as a (contour) saliency map. Finally, they align the UCMs obtained at different resolutions into a single UCM.

In [60], the authors propose a hierarchical segmentation method using Convolutional Neural Networks (CNN). Each level of a CNN conveys information regarding different levels of resolution of an image. Hence, the authors use the output of each level of a CNN to compute oriented boundaries and, subsequently, UCMs. The UCMs obtained at different levels are further aligned using a faster implementation of the method proposed in [7], producing hierarchies with state-of-the-art performance in several computer vision applications.

With so many algorithms to compute hierarchies of partitions, it became necessary to evaluate the contribution of each hierarchy with respect to different tasks [5, 83, 82, 79]. Usually, large annotated image datasets are used to evaluate hierarchical segmentation algorithms. Those evaluations are often empirical in the sense that an algorithm is evaluated with respect to its output on a set of images. As the manual annotations provided by large image datasets [63, 16, 1, 71] are not hierarchical, hierarchies of partitions are commonly evaluated by comparing each segmentation of the hierarchy against the image ground truth.

Aiming at expanding the search space of image segmentation problems, the notion of hierarchies of partitions is extended to braids of partitions in [53]. A braid of partitions is composed of partitions that locally follow the causality and location principles: given any two partitions $\mathbf{P}_1$ and $\mathbf{P}_2$ of a braid of partitions, every region of $\mathbf{P}_1$ is either a subset of a region of $\mathbf{P}_2$ or it is composed of regions of $\mathbf{P}_2$. Hence, any hierarchy of partitions is a braid of partitions but the other implication is not true in general.

Several well-known image segmentation techniques are modeled in the framework of graphs [17, 90, 35, 31, 43, 21], including (hierarchical) watersheds [66, 22, 24, 27, 74]. In this thesis, we focus on morphological hierarchies built in the framework of weighted graphs and, in particular, on hierarchical watersheds and on their link with other morphological hierarchies such as the binary partition trees.

2.3 Quasi-flat zones hierarchies

In this section, we first present the definition of quasi-flat zone hierarchy in the framework of weighted graphs. Then, we present some of the applications of quasi-flat zones in the context of image segmentation.

2.3.1 Notations and definitions

Let (G, w) be a weighted graph and let λ be any element in $\mathbb{R}$. Let V and E be the vertex and edge sets of G, respectively. The *λ-level set of* (G, w) is the graph $(V, E_\lambda(G))$ such that $E_\lambda(G) = \{u \in E(G) \mid w(u) \leq \lambda\}$. Without loss of generality, let us assume that the range of w is included in the set $\mathbb{E}$ of all integers from 0 to $|E| - 1$ (otherwise, one could always consider an increasing one-to-one correspondence from the set $\{w(u) \mid u \in E\}$ into the subset $\{0, ..., |\{w(u) \mid u \in E\}| - 1\}$ of $\mathbb{E}$). The sequence

$$\mathcal{QFZ}(w) = (CC(G_{\lambda,w}) \mid \lambda \in \mathbb{E}) \tag{2.1}$$

where $G_{\lambda,w}$ is the λ-level set of (G, w), is a hierarchy called the *Quasi-Flat Zones (QFZ) hierarchy (of w)* [72, 69, 92, 27].

For instance, the hierarchy $\mathcal{H}$ of Figure 2.1(a) is the QFZ hierarchy of the graph (G, w) of Figure 2.1(b).

2.3.2 Quasi-flat zones for image segmentation

Let I be a gray-scale image such that the gray value of any pixel of I is in the range $[0, 255]$. A flat-zone of I is a set of connected pixels (*e.g.* 4 or 8 connected pixels) with uniform gray-level. In the context of edge weighted graphs, a flat zone is a set of vertices linked by edges with uniform weight. Let k be a value in $[0, 255]$. The k-level set of I is the set of pixels of I with gray-levels greater than k. Any gray-scale image can be equally represented and reconstructed from its level sets.

Connected operators act on the connected components of the level sets of an image (or on the complement of the level sets of an image). Hence, connected operators filter out or merge connected components of the level sets of an image without creating new contours, which is very useful for image simplification. As mentioned in Section 2.2.3, connected operators can be implemented through hierarchical representations of an image, including min-tree, max-tree and tree of shapes.

A quasi-flat zone of an image is a largest set of connected pixels such that the difference in gray-level between two neighbour pixels is limited by a given threshold k. For any k, we can define a partition of the image. By stacking the partitions of quasi-flat zones of an image for increasing values of k, we obtain a sequence of partitions for which the causality and location properties hold true, resulting in the quasi-flat zones hierarchy. From this definition, we can infer two features of quasi-flat zones partitions. First, quasi-flat zones partitions are prone to connect dissimilar regions that are linked only by a narrow path (leakage problem). Second, dissimilar regions can be connected by a path in which gray-levels vary smoothly (chaining problem).

The first use of quasi-flat zones dates back to [72] in the analysis of aerial photographs. In order to segment and classify regions of aerial images into forest, cropts, houses, etc., the authors use the quasi-flat zones of the smoothed images for a given threshold, along with colorimetric and geometric information.

In [92], the author addresses the chaining problem of quasi-flat zones, where pixels of large gray-level difference are connected by a path of low gray-level variation between neighbouring pixels. As a solution to this issue, he proposes the introduction of a new parameter to control the maximal gray-level variation between pixels of a same region. They denote quasi-flat zones as α-connected components.

In [69], the authors introduce a morphological scale space representation of images based on the notion of levelings. A function g is a leveling of a function f if, for any two neighbouring pixels x and y, we have that $g(x) > g(y)$ implies that $f(x) \geq g(x)$ and $g(y) \geq g(x)$. They prove that levelings are connected filters, hence the link with quasi-flat zones. They show that, as quasi-flat zones, levelings with increasing parameters lead to a hierarchical representation of an image obeying the causality and location principle of the regions and contours.

In [27], the authors link quasi-flat zones hierarchies with other morphological hierarchies in the context of weighted graphs. They show that a quasi-flat zone hierarchy can be obtained by simplifying a binary partition tree. In particular, quasi-flat zones hierarchies are linked to saliency maps, which are a compact representation of hierarchies of partitions, as discussed in the next section.

2.4 Contour saliency maps

Until now, we have considered hierarchies of partitions represented by the inclusion relationship between regions or by a sequence of partitions. In this section, we introduce a dual representation of hierarchies of partitions. Instead of a sequence of partitions, we characterize a hierarchy of partitions by the contours between the regions of each partition. As established in [25], a connected hierarchy can be equivalently treated by means of a weighted graph through the notion of a (contour) saliency map (also known as ultrametric contour map [5]). A saliency map is a map from the contours present in the partitions of a hierarchy into a set of values indicating the level of disappearance of each contour. Through the definition of quasi-flat zones, any hierarchy can be recovered from its saliency map.

Saliency maps and hierarchies are closely related to the notion of ultrametric distances [73, 5]. An ultrametric distance is a metric space that satisfies the ultrametric inequality: given an ultrametric distance map d and three points x, y and z in the space, we have $d(x, y) \leq max\{d(x, z), d(z, y)\}$. Given a hierarchy $\mathcal{H} = (\mathbf{P}_0, \ldots, \mathbf{P}_\ell)$ on a set V, let f be a function from $V \times V$ into the set $\{0, \ldots, \ell\}$ such that, for any two vertices x and y in V, $f(x, y)$ is the lowest k such that x and y belong to the same region of $\mathbf{P}_k$. We can observe that:

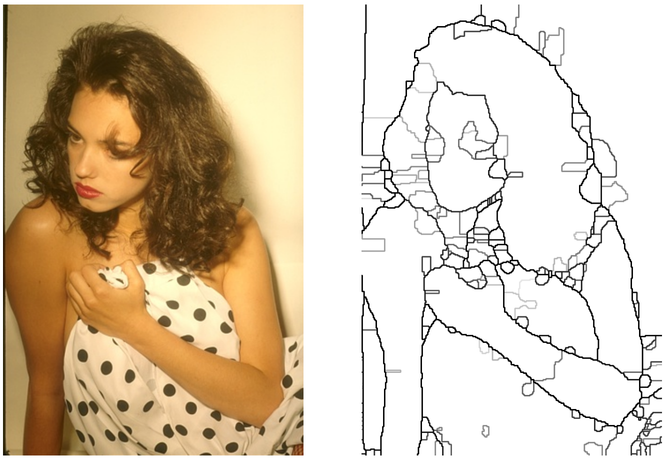
- for any x in V, $f(x, x) = 0$; and
- for any x and y in V, $f(x, y) \geq 0$ and $f(x, y) = f(y, x)$; and
- for any x, y, z in V, $f(x, y) \leq max\{f(x, z), f(z, y)\}$ because the level in which x and y belong to the same region is necessarily finer than the level in which x, y and z belong to the same region.

Hence, f is an ultrametric distance and we can see that the hierarchy $\mathcal{H}$ can be recovered from f. Indeed, if $\mathcal{H}$ is connected for a given graph G, the values of f for the edges of G suffice to recover the hierarchy $\mathcal{H}$. This map f' from the edges of E into their value in f is the saliency map of $\mathcal{H}$. The reader may note that, in other contexts, a saliency map denotes a map that highlights the objects of interest of an image (high values for pixels belonging to important regions), which is not our case. Here, the saliency values are assigned to the contours and not to the interior of the regions.

The first definition of saliency maps in the context of hierarchies of partitions was presented by Najman and Schmitt [75]. In [75], the authors extend the definition of dynamics of minima [45] to dynamics of contours. They define a map from each contour of a watershed segmentation into the saliency of this contour, where any threshold of the resulting map produces closed contours.

In this work, we focus on connected hierarchies. Let G be a graph and let $\mathcal{H}$ be a hierarchy connected for G. The saliency map of $\mathcal{H}$ is defined on the edges of G because all vertices of G belong to a region of $\mathcal{H}$. In this context, saliency maps are represented thanks to cubical complexes [26]: the representation of a saliency map is an image with the double number of lines and columns of the original image and where every vertex and every edge is represented by a pixel.

In Figure 2.4, we show a representation of a saliency map. In this representation, the darkest contours are the ones that persist at the highest levels of the hierarchy.



(a) I (b) f

Figure 2.4: An image I and the saliency map f of a hierarchy of partitions of I obtained with the method proposed in [60].

2.4.1 Notations and definitions

Let $G = (V, E)$ be a graph. Given a hierarchy $\mathcal{H} = (\mathbf{P}_0, \ldots, \mathbf{P}_\ell)$ which is connected for G, the *(contour) saliency map of* $\mathcal{H}$ is the map from E into $\{0, \ldots, \ell\}$, denoted by $\Phi(\mathcal{H})$, such that, for any edge $u = \{x, y\}$ in E, the value $\Phi(\mathcal{H})(u)$ is the lowest value i in $\{0, \ldots, \ell\}$ such that x and y belong to a same region of $\mathbf{P}_i$. It follows that any connected hierarchy has a unique saliency map. Moreover, any hierarchy $\mathcal{H}$ connected for G is precisely the quasi-flat zones hierarchy of its own saliency map: $\mathcal{H} = \mathcal{QFZ}(\Phi(\mathcal{H}))$.

For instance, the map depicted in Figure 2.1(b) is the saliency map of the hierarchy of Figure 2.1(a).

Let $G = (V, E)$ be a graph and let $\mathcal{H} = (\mathbf{P}_1, \ldots, \mathbf{P}_\ell)$ be a hierarchy on V. Let d be a map from $V \times V$ into $\mathbb{R}$ such that, for any pair (x, y) of vertices in $V \times V$, the value $d(x, y)$ is the greatest edge weight λ in a path π from x to y (resp. y to x) in $(G, \Phi(\mathcal{H}))$ and such that, for any other path π' from x to y (resp. y to x), the greatest edge weight in π' is greater than or equal to λ. For any egde $u = \{x, y\}$ in E, we say that $d(x, y)$ *is the ultrametric distance between* x *and* y *in* $(G, \Phi(\mathcal{H}))$. We can affirm that (V, d) is an ultrametric space. Moreover, for any two vertices x and y in V, by the definition of saliency maps and considering its link with QFZ hierarchies, we may say that $d(x, y)$ is the lowest value λ such that x and y belong to a same region of the partition $\mathbf{P}_\lambda$ of $\mathcal{H}$. Furthermore, if G is a complete graph, we can conclude that $(V, \Phi(\mathcal{H}))$ is an ultrametric space.

2.5 Binary partition trees

Binary partition trees [87] are widely used for hierarchical image representation. In this section, we first review the definition of binary partition tree and some of its applications. Then, we describe the particular case where the merging order is defined by the edge weights [27]. As we will see along this manuscript, the latter is deeply connected to hierarchical watersheds [27] and can be used to study properties of hierarchical watersheds.

2.5.1 Introduction

In [87], the notion of binary partition tree (BPT) is introduced aiming to fuse the flexibility of the order in which regions are merged by segmentation algorithms and the flexibility offered by connected operators in the processing of the max-tree.

The execution of segmentation algorithms based on merging criteria involves three concepts: merging order, merging criteria and region model. Given an initial set of regions or superpixels, the merging order determines the order in which pairs of neighbouring regions should be considered for merging, which is given by the similarity between regions according to a given criterion, *e.g.* average gray level. The merging criterion determines when the merging process stops, *e.g.* when a given number of regions is reached. The region model determines how each region is represented after each merging step, *e.g.* average gray-level of the pixels in a region.

As discussed in Section 2.3, connected operators are operators that act on the connected components of the flat-zones of thresholded versions of an image. Given an image I and a connected operator Ψ, we can say that the partition induced by the flat-zones of $\Psi(I)$ at level λ is coarser than the partition induced by the flat-zones of I. Connected operators can be efficiently implemented through max-trees: given a max-tree T, Ψ is a filtering of the regions of T according to a given criterion. When this criterion is increasing on the nodes of T, Ψ simply filters out all descendants of any region that does not follow the criterion. Otherwise, if the criterion is non increasing, the result is not robust in the sense that similar images can have different results. Strategies to handle non-increasing criteria are discussed [88].

A BPT as presented in [87] can be constructed through algorithms based on a merging criterion. The BPT is obtained by keeping track of the merging sequence of an algorithm, *i.e.*, the sequence of pair of regions of a segmentation that are merged by a merging algorithm. The merging criterion is the merging of all initial regions into a single region. Then, the processing of a BPT follows the same pruning strategies used in the processing of the max-tree by connected operators.

Among the applications of the BPT, Salembier and Garrido [87] highlight:

1. detection and recognition of regions based on a given criteria *e.g.* circularity. The BPT offers a set of 2N-1 regions (where N is the number of initial regions), which limits the search space to a small number of reasonably homogeneous regions;

2. image compression for low bandwidth servers. Instead of sending the color information of each pixel individually, some regions of the image can be sent as a superpixel (only the contours and a constant color are sent). The distortion of the resulting image and the budget to send an image can be optimized on the BPT when both distortion and budged are increasing on the regions of the BPT; and

3. image segmentation. Image segmentations can be extracted from the BPT by simply following the merging order used to construct the BPT. The desired number of regions can be used as a merging criterion and the final segmentation can be obtained by filtering the BPT as done by connected operators on the max-tree. This latter approach is called direct segmentation. Alternatively, segmentations can be obtained by propagating markers from the leaves, at the pixel level, to the root. This marked segmentation technique has been notably used in the evaluation framework of hierarchies proposed in [80, 79] to be discussed in Chapter 6.

BPTs have been largely used in remote sensing image processing [99, 11, 86]. The advance in this area, leading to larger scale images, call for a method to efficiently simplify an image and to decrease the search space of the objects in a remote sensing image.

In [99], the authors segment hyperspectral images by applying a Support Vector Machine (SVM) to nodes of a BPT. Given a BPT computed from a hyperspectral image, the nodes of this BPT are classified by an SVM according to their impurity level, which is related to the number of different classes assigned to the descendants of a node. Then, this BPT is pruned and the class of each pixel is determined by the leaf region that contains this pixel. The authors show that a simple SVM classification is improved with the aid of a BPT.

In [86], the authors proposed a multi-criteria and multi-image binary partition tree computation with application to remote sensing image segmentation. They build a BPT from several photographs of the same scene. Then, they work on a set of images with the same dimensions, where each image induces a different dissimilarity graph (gradient) and the merging criterion is defined by alternating the information provided by each graph.

In [100], BPT is used for object detection with application to face and traffic sign detection. Using a merging criterion based on color and contour complexity, the authors study methods to obtain the initial partition of the BPT and the merging sequence separately.

In [27], the authors establish the link between BPTs, minimum spanning tree and other morphological hierarchies, in particular the min-trees, quasi-flat zones hierarchies and hierarchical watersheds. They introduce a particular case of the BPTs denoted by binary partition hierarchy by altitude ordering (BPHAO), which can be used to obtain QFZ hierarchies, min-trees and hierarchical watersheds. The link between BPHAO and hierarchical watersheds are the basis of our research on characterization of hierarchical watersheds (see Chapter 3), on the watersheding operator (see Chapter 4) and on probabilities of hierarchical watersheds (Chapter 5).

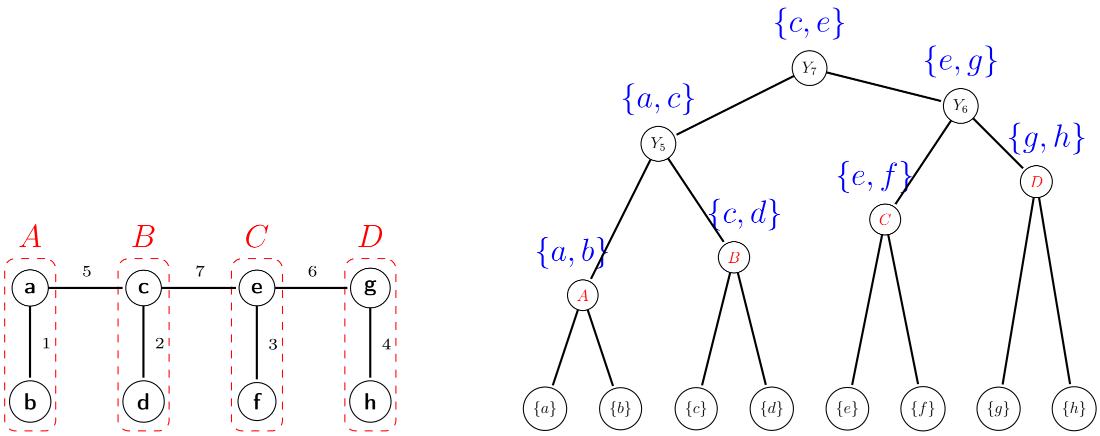
BPHAOS are deeply related to single linkage clustering [42]. Hence, the link between MST and single linkage clustering established in [42] can be extended to BPHAOs. For example, MST algorithms have been successfully used by Najman et al. [74] to compute BPHAOs. The authors of [74] also provide an efficient post-processing of the BPT to find the minima and watershed-cut edges of a graph, as explained in Section 2.6.1.

In the next section, we formalize the definition of BPHAO in the framework of weighted graphs.

2.5.2 Notations and definitions

Let (G, w) be a weighted graph. Let V and E be the vertex and edge sets of G, respectively. Let $\prec$ be a total ordering (on E), *i.e.*, $\prec$ is a binary relation that is transitive and trichotomous: for any u and v in E only one of the relations $u \prec v$, $v \prec u$ and $v = u$ holds true. We say that *$\prec$ is an altitude ordering (on E) for w* if, for any u and v in E, if $w(u) < w(v)$, then $u \prec v$. Let $\prec$ be an altitude ordering for w. Let k be any element in $\{1, \ldots, |E|\}$. We denote by $u_k^\prec$ the k-th element of E with respect to $\prec$. We set $\mathbf{B}_0 = \{\{x\} \mid x \in V\}$. The *$k$-partition of V (by the ordering $\prec$)* is defined by $\mathbf{B}_k = \{\mathbf{B}_{k-1}^y \cup \mathbf{B}_{k-1}^x\} \cup (\mathbf{B}_{k-1} \setminus \{\mathbf{B}_{k-1}^x, \mathbf{B}_{k-1}^y\})$ where $u_k^\prec = \{x, y\}$ and $\mathbf{B}_{k-1}^x$ and $\mathbf{B}_{k-1}^y$ are the regions of $\mathbf{B}_{k-1}$ that contain x and y, respectively. The sequence $(\mathbf{B}_i \mid i = 0 \textit{ or } \mathbf{B}_i \neq \mathbf{B}_{i-1})$ is a hierarchy on V. This hierarchy $(\mathbf{B}_i \mid i = 0 \textit{ or } \mathbf{B}_i \neq \mathbf{B}_{i-1})$, denoted by $\mathcal{B}_\prec$, is called the *binary partition hierarchy (by altitude ordering) of (G, w) by $\prec$*.

Let $\mathcal{B}$ be a hierarchy on V. We say that *$\mathcal{B}$ is a binary partition hierarchy (by altitude ordering) of* (G, w) if there is an altitude ordering $\prec$ for w such that $\mathcal{B}$ is the binary partition hierarchy of (G, w) by $\prec$.



(a) (G, w) (b) $\mathcal{B}$

Figure 2.5: (a): A weighted graph (G, w) with four minima delimited by the dashed lines. (b): The unique binary partition hierarchy $\mathcal{B}$ of (G, w).

Let $\prec$ be an altitude ordering for w. We can associate any non-leaf region X of the binary partition hierarchy $\mathcal{B}_\prec$ of (G, w) by $\prec$ to the lowest rank r such that $\mathbf{B}_r$ contains X. This rank is called the *rank of* X. Let X be a non-leaf region of $\mathcal{B}_\prec$ and let r be the rank of X. The *building edge of* X is the r-th edge for $\prec$. Given an edge u in E, if u is the building edge of a region of $\mathcal{B}_\prec$, we say that u *is a building edge for* $\prec$. Given a building edge u for $\prec$, we denote the region of $\mathcal{B}_\prec$ whose building edge is u by R_u. The set of all building edges for $\prec$ is denoted by $E_\prec$.

Let (G, w) be the weighted graph illustrated in Figure 2.5(a) and let $\mathcal{B}$ be the binary partition hierarchy of (G, w) illustrated in Figure 2.5(b). We can see that $\mathcal{B}$ is the binary partition hierarchy of (G, w) by the altitude ordering $\prec$ such that $\{a, b\} \prec \{c, d\} \prec \{e, f\} \prec \{g, h\} \prec \{a, c\} \prec \{e, g\} \prec \{c, e\}$. The building edge of each non-leaf region R of $\mathcal{B}$ is shown above the node that represents R.

Let $\mathcal{B}$ be a binary partition hierarchy of (G, w) and let X and Y be two distinct regions of $\mathcal{B}$. If the parent of X is equal to the parent of Y, we say that X is a sibling of Y, that Y is a sibling of X and that X and Y are siblings. It can be seen that any region $R \neq V$ of $\mathcal{B}$ has exactly one sibling and we denote this unique sibling of R by $sibling(R)$.

2.6 Hierarchical watersheds

In this section, we first present the notations and definitions related to hierarchical watersheds in the sense of minimum spanning forests [24, 27], and the link between binary partitions hierarchies by altitude ordering (see Section 2.5.2) and the minima and watershed-cut edges of a graph. Subsequently, we formalize the notion of marker-based segmentation, and we compare watersheds with other common energy terms used in graph based segmentation. Finally, we review the watershed transform and watershed based hierarchies in the context of graphs and image segmentation.

2.6.1 Notations and definitions

Let (G, w) be a connected weighted graph. Let k be a value in $\mathbb{R}$. As established in Section 2.1, a connected subgraph G' of G is a minimum (of w) at level k if:

1. $E(G') \neq \emptyset$; and

2. for any edge u in $E(G')$, the weight of u is equal to k; and

3. for any edge $\{x, y\}$ in $E \setminus E(G')$ such that $|\{x, y\} \cap V(G')| \geq 1$, the weight of $\{x, y\}$ is strictly greater than k.

In the remainder of this section, let (G, w) be a connected weighted graph and let n be the number of minima of w.

Let $\{G_1, \ldots, G_\ell\}$ be a set of graphs. We denote by $\sqcup\{G_1, \ldots, G_\ell\}$ the graph $(\cup\{V(G_j) \mid j \in \{1, \ldots, \ell\}\}, \cup\{E(G_j) \mid j \in \{1, \ldots, \ell\}\})$. In the following, we define hierarchical watersheds based on minimum spanning forests following the definition of [24, 27].

Definition 1 (hierarchical watershed [24, 27]). *Let $\mathcal{S} = (M_1, \ldots, M_n)$ be a sequence of n pairwise distinct minima of w. Let $(G_0, \ldots, G_{n-1})$ be a sequence of subgraphs of G such that:*

1. *for any i in $\{0, \ldots, n-1\}$, the graph G_i is a MSF of G rooted in $\sqcup\{M_j \mid j \in \{i+1, \ldots, n\}\}$; and*

2. *for any i in $\{1, \ldots, n-1\}$, G_{i-1} is a subgraph of G_i.*

The sequence $\mathcal{T} = (CC(G_0), \ldots, CC(G_{n-1}))$ is called a hierarchical watershed *of (G, w) for $\mathcal{S}$. Given a hierarchy $\mathcal{H}$, we say that $\mathcal{H}$ is a* hierarchical watershed of (G, w) *if there exists a sequence $\mathcal{S}$ of minima of w such that $\mathcal{H}$ is a hierarchical watershed of (G, w) for $\mathcal{S}$.*

For instance, let (G, w) and $\mathcal{H}$ be the weighted graph and the hierarchy shown in Figure 2.6(a) and (b), respectively. We can see that $\mathcal{H}$ is the hierarchical watershed of (G, w) for the sequence (C, A, B, D) of minima of w.

Important notation: by abuse of terminology, when no confusion is possible, if M is a minimum of w, we call the set $V(M)$ of vertices of M as a minimum of w.

As established in [74, 27], hierarchical watersheds are deeply linked to the notion of binary partition hierarchy (by altitude ordering). Using the algorithm proposed in [74], a binary partition hierarchy of (G, w) can be constructed by iteratively merging pairs of regions (subsets of V) connected by edges of increasing weights. Given any minimum M of w, each vertex x of M is connected to another vertex in M by an edge of lower weight than the edges linking x to other vertices that do not belong to M. Hence, by considering any altitude ordering for w, the set of vertices of M are merged before any other vertices are merged with M. Therefore, every minimum of w is a region of any binary partition hierarchy of (G, w).

In Cousty *et al.* [22], the authors formalize the notion of watershed-cuts in edge-weighted graphs, and establish the link between watershed-cuts and MSFs. In this context, a watershed-cut edge of (G, w) is an edge that links distinct "catchment basins", *i.e.*, connected components of the MSF rooted in the minima of w. In fact, since there may be several MSFs rooted in the minima of w, the set of watershed-cut egdes of (G, w) is not unique. Indeed, for a fixed altitude ordering for w, we can define a unique set of watershed-cut edges of (G, w). As established in [74], given an altitude ordering $\prec$, the watershed-cut edges (for $\prec$), whose definition is given bellow, can be obtained from the binary partition hierarchy by $\prec$ along with the minima of w.

Definition 2 (watershed-cut edge for an altitude ordering)**.** *Let $\prec$ be an altitude ordering for w and let u be a building edge for $\prec$. We say that* u is a watershed-cut edge (of (G, w)) for $\prec$ *if each child of the region R_u of $\mathcal{B}_\prec$ includes at least one minimum of w.*

The link between binary partition hierarchies and hierarchical watersheds provided in [74, 27] induce an efficient method to obtain the saliency map of a hierarchical watershed, which is connected to the theoretical results introduced in this thesis. In order to revise the method to compute the saliency map of hierarchical watersheds proposed in [74, 27], we present the definitions of extinction values and persistence values for a sequence of minima, and the notion of a hierarchy induced by an altitude ordering and by a sequence of minima.

Definition 3 (extinction value for a sequence of minima [27])**.** *Let (G, w) be an edge weighted graph. Let $\prec$ be an altitude ordering for w and let $\mathcal{B}_\prec$ be the binary partition* *hierarchy by $\prec$, as defined in Section 2.5.2. Let $\mathcal{S} = (M_1, \ldots, M_n)$ be a sequence of pairwise distinct minima of w and let R be a region of $\mathcal{B}_\prec$. The* extinction value of R for $(\mathcal{S},\prec)$ *is zero if there is no minimum of w included in R and, otherwise, it is the maximum value i in $\{1, \ldots, n\}$ such that the minimum M_i is included in R.*

Definition 4 (persistence value for a sequence of minima [27])**.** *Let (G, w) be a weighted graph. Let $\prec$ be an altitude ordering for w, let $\mathcal{B}_\prec$ be the binary partition hierarchy by $\prec$ and let $\mathcal{S}$ be a sequence of minima of w. Let u be a building edge for $\prec$ and let X be the region of $\mathcal{B}_\prec$ whose building edge is u. The persistence value of u for $(\mathcal{S}, \prec)$ is the minimum of the extinction values of the children of X.*

Definition 5. (hierarchy induced by an altitude ordering and by a sequence of minima [27]) *Let (G, w) be a weighted graph and let n be the number of minima of w. Let $\prec$ be an altitude ordering for w and let $\mathcal{S}$ be a sequence of minima of w. Let ρ be the map from the building edges (for $\prec$) into $\mathbb{R}$ such that, for any building edge u, $\rho(u)$ is the persistence value of u for $(\mathcal{S}, \prec)$. Let B_i the set of building edges of $\mathcal{B}_\prec$ whose persistence value is lower than or equal to i. The sequence of partitions $(CC(V, B_0), \ldots, CC(V, B_{n-1}))$ is a hierarchy called* the hierarchy induced by $\prec$ and $\mathcal{S}$.

From the results of [27], we state the following property.

Property 6 (Property 12 of [27])**.** *Let (G, w) be a weighted graph. Let $\mathcal{S}$ be a sequence of minima of w and let $\mathcal{H}$ be a hierarchy on G. The hierarchy $\mathcal{H}$ is a hierarchical watershed of (G, w) for $\mathcal{S}$ if and only if there exists an altitude ordering for w such that $\mathcal{H}$ is the hierarchy induced by $\prec$ and $\mathcal{S}$.*

Given a graph (G, w) and a sequence $\mathcal{S}$ of minima of w, by Property 6, the saliency map of a hierarchical watershed of (G, w) for $\mathcal{S}$ can be obtained through the following steps:

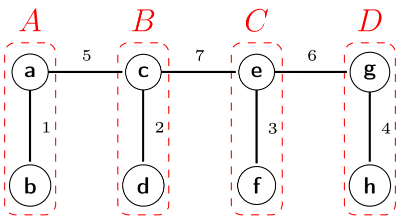
1. computation of an altitude ordering $\prec$ for w;

2. computation of the binary partition hierarchy $\mathcal{B}_\prec$ by $\prec$;

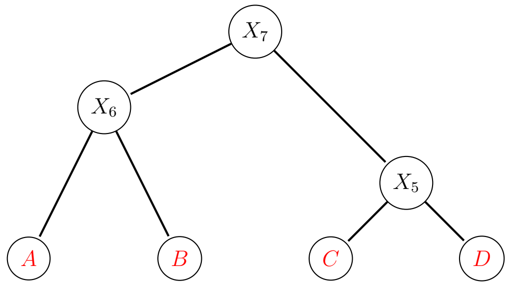
3. computation of the extinction values for $(\mathcal{S},\prec)$;

4. computation of the persistence values for $(\mathcal{S}, \prec)$; and

5. computation of the saliency map f of a hierarchical watershed of (G, w) for $\mathcal{S}$. Given the set $E_\prec$ of building edges for $\prec$, let ρ be the map from $E_\prec$ into $\mathbb{R}$ such that, for any edge u in $E_\prec$, the value $\rho(u)$ is the persistence value of u for $(\mathcal{S}, \prec)$.



(a) (G, w)



(b) $\mathcal{H}$

Figure 2.6: (a): A weighted graph (G, w) with four minima delimited by the dashed lines. (b): The hierarchical watershed $\mathcal{H}$ of (G, w) for the sequence (C, A, B, D) of minima of w.

For every edge $u = \{x, y\}$ of G, the value $f(u)$ is the ultrametric distance between x and y in $((V(G), E_{\prec}), \rho)$.

Let us consider that the edges of G are already ordered or can be ordered in linear time. In this case, the first three steps can be executed in quasi-linear times with respect to the number of edges of G, as established in [74]. Moreover, if G is a tree, the map ρ is precisely the saliency map of a hierarchical watershed of (G, w) for $\mathcal{S}$ and, consequently, this saliency map can be obtained in quasi-linear time as well.

2.6.2 Marker-based segmentation

The hierarchical watersheds (see Definition 1) are the solutions to a multiscale optimization problem, namely each partition of a hierarchical watershed optimizes a simple cost function. More precisely, given a graph (G, w), each partition of a hierarchical watershed of (G, w) is induced by a solution to the problem of finding a MSF of (G, w) rooted in a certain subset of the minima of w. Moreover, the (optimal) partitions of a hierarchical watershed satisfy a hierarchical or scale consistency property formalized below in the context of marker-based segmentation. However, as we will see in this section, this property is not satisfied by some of the most common energy terms used in graph based image segmentation.

In the remainder of this section, let us assume that (G, w) is a connected weighted graph. A *marker set (of (G, w))* is a set of disjoint subsets of V. We denote by Π_V the set of all partitions of V.

Definition 7 (marker-based segmentation)**.** *Let $\mathcal{M} = \{M_1, \ldots, M_\ell\}$ be a marker set. A* marker-based segmentation of (G, w) for $\mathcal{M}$ *is a partition $\mathbf{P}$ in Π_V such that each region of $\mathbf{P}$ includes exactly one element of $\mathcal{M}$.*

Marker-based segmentations can be obtained by watershed [22], min-cut [94], average-cut [102] and shortest path forest [23] algorithms, to name a few. It can be observed that the related optimization problems are ill-posed and do not necessarily have a unique solution. Therefore, those algorithms are not deterministic: they can produce several solutions for a given marker set. Hence, in order to study the "hierarchical behavior" of these algorithms, we start by providing a definition of a non-deterministic marker-based segmentation operator.

Definition 8. *A* (non-deterministic) marker-based segmentation operator σ *is a map from the set of all marker sets into the set of all subsets of* Π_V *such that, for any marker set* $\mathcal{M}$*, any partition in* $\sigma(\mathcal{M})$ *is a marker-based segmentation for* $\mathcal{M}$*.*

Definition 9. *Let* σ *be a marker-based segmentation operator. We say that* σ is hierarchical *if, for any two marker sets* $\mathcal{M}$ *and* $\mathcal{M}'$ *such that* $\mathcal{M}'$ *is a subset of* $\mathcal{M}$*, there exists a pair* $(\mathbf{P}, \mathbf{P}')$ *of partitions such that* $\mathbf{P}$ *and* $\mathbf{P}'$ *belongs to respectively* $\sigma(\mathcal{M})$ *and* $\sigma(\mathcal{M}')$*, and such that* $\mathbf{P}$ *is a refinement of* $\mathbf{P}'$*.*

Let L be a subset of E. We say that *L is a cut if*, for any edge $u = \{x, y\}$ in L, x and y belong to distinct connected components of $(V, E \setminus L)$. We denote the set of connected components of $(V, E \setminus L)$ by the *partition induced by L*. By abuse of notation, given a spanning forest G' of G, we also denote the set of connected components of G' by the *partition induced by G'*.

The following property asserts that the MSF operator is indeed hierarchical.

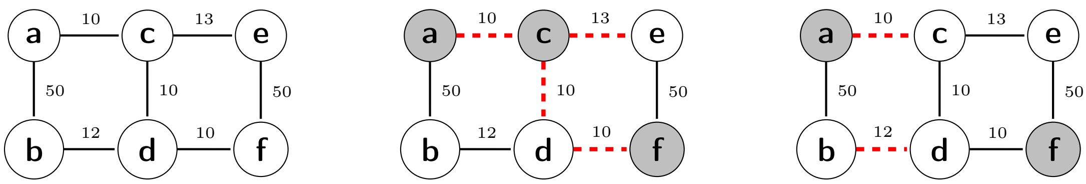
Property 10. *Let* σ *be the operator that maps any marker set* $\mathcal{M}$ *into the set of partitions induced by each of the MSFs rooted in* $\mathcal{M}$*. Then, the operator* σ *is hierarchical.*

Property 10 is the basis of hierarchical watersheds (Definition 1). In the following, we show that the operators that produce min-cuts, average-cuts and shortest path forest cuts are not hierarchical.

Definition 11 (min-cuts)**.** *Let* $\mathcal{M}$ *be a marker set. A* min-cut of (G, w) for $\mathcal{M}$ *is a subset* L *of* E *such that:*

1. *the set of connected components of* $(V, E \setminus L)$ *is a marker-based segmentation of* (G, w) *for* $\mathcal{M}$*; and*
2. *the sum* $\sum_{u \in L} w(u)$ *is minimal for all subsets of* E *for which statement 1 holds true.*

Property 12. *Let* σ_{min} *be the operator that maps any marker set* $\mathcal{M}$ *into the set of partitions induced by each of the min-cuts of* (G, w) *for* $\mathcal{M}$*. The operator* σ_{min} *is not hierarchical.*



(a) (G, w) (b) L (c) L'

Figure 2.7: From left to right: a graph (G, w), the min-cut L (dashed edges) of (G, w) for the set of markers $\mathcal{M} = \{\{a\}, \{c\}, \{f\}\}$, and the min-cut L' of (G, w) for the set of markers $\mathcal{M}' = \{\{a\}, \{f\}\}$. The partition induced by L is not a refinement of the partition induced by L'.

Proof. Let (G, w) be the weighted graph of Figure 2.7(a). Let $\mathcal{M} = \{\{a\}, \{c\}, \{f\}\}$ be a marker set of (G, w). In Figure 2.7(b), we show the unique min-cut L (dashed edges) of (G, w) for $\mathcal{M}$. Hence, we have $\sigma_{min}(\mathcal{M}) = \{\{\{a, b, d\}, \{c\}, \{e, f\}\}\}$. In Figure 2.7(c), we shown the unique min-cut L' of (G, w) for the subset $\mathcal{M}' = \{\{a\}, \{f\}\}$ of $\mathcal{M}$. Therefore, we have $\sigma_{min}(\mathcal{M}') = \{\{\{a, b\}, \{c, d, e, f\}\}\}$. We can observe that the unique partition in $\sigma_{min}(\mathcal{M})$ is not a refinement of the unique partition in $\sigma_{min}(\mathcal{M}')$. Thus, σ_{min} is not hierarchical.

□

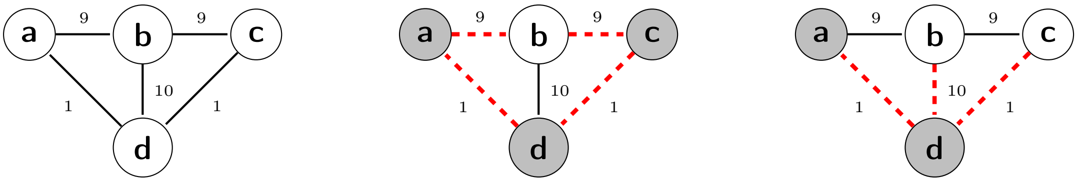
Definition 13 (average-cuts)**.** *Let $\mathcal{M}$ be a marker set. An* average-cut of (G, w) for $\mathcal{M}$ *is a subset L of E such that:*

1. *the set of connected components of $(V, E \setminus L)$ is a marker-based segmentation of (G, w) for $\mathcal{M}$; and*
2. *the value $\frac{\sum_{u \in L} w(u)}{|L|}$ is minimal for all subsets of E for which statement 1 holds true.*

Property 14. *Let σ_{avg} be the operator that maps any marker set $\mathcal{M}$ into the set of partitions induced by each of the average-cuts of (G, w) for $\mathcal{M}$. The operator σ_{avg} is not hierarchical.*

Proof. Let (G, w) be the weighted graph of Figure 2.8(a). Let $\mathcal{M} = \{\{a\}, \{c\}, \{d\}\}$ be a marker set of (G, w). In Figure 2.8(b), we show the unique average-cut L (dashed edges) of (G, w) for $\mathcal{M}$. Hence, we have $\sigma_{avg}(\mathcal{M}) = \{\{\{a\}, \{b, d\}, \{c\}\}\}$. In Figure 2.8(c), we shown the unique average-cut L' of (G, w) for the subset $\mathcal{M}' = \{\{a\}, \{d\}\}$ of $\mathcal{M}$. Therefore, we have $\sigma_{avg}(\mathcal{M}') = \{\{\{a, b, c\}, \{d\}\}\}$. We can observe that the unique partition in $\sigma_{avg}(\mathcal{M})$ is not a refinement of the unique partition in $\sigma_{avg}(\mathcal{M}')$. Thus, σ_{avg} is not hierarchical.

□



(a) (G, w) (b) L (c) L'

Figure 2.8: From left to right: a graph (G, w), the average-cut L (dashed edges) of (G, w) for the set of markers $\mathcal{M} = \{\{a\}, \{c\}, \{d\}\}$, and the average-cut L' of (G, w) for the set of markers $\mathcal{M}' = \{\{a\}, \{d\}\}$. The partition induced by L is not a refinement of the partition induced by L'.

Let d be a distance on V, *i.e.*, a map from $V \times V$ to $\mathbb{R}^+$ such that:

- for any two vertices x and y in V, $d(x, y) = d(y, x)$;
- for any two vertices x and y in V, $d(x, y) = 0$ if and only if $x = y$; and
- for any three vertices x, y and z, we have $d(x, y) \leq d(x, z) + d(z, y)$.

Let $\mathcal{M}$ be a marker set and let x be a vertex in V. Let π be a path from x to y such that y belongs to an element of $\mathcal{M}$. We say that *π is a d-shortest path from x to $\mathcal{M}$* if the distance $d(x, y)$ is less than the distance $d(x, z)$ for any other vertex z such that z belongs to an element of $\mathcal{M}$.

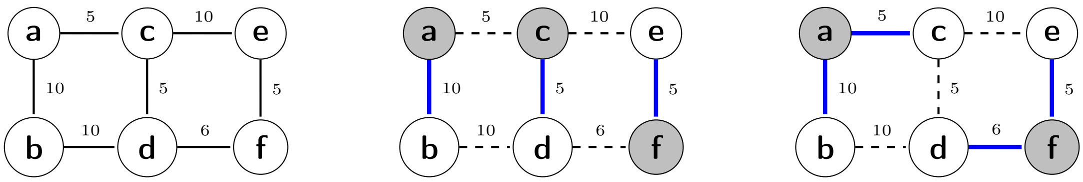
Definition 15 (shortest path forests)**.** *Let $\mathcal{M}$ be a marker set and let d be a distance on V. Let G' be a forest of (G, w) rooted in $\mathcal{M}$. The graph G'* is a d-shortest path forest of (G, w) for $\mathcal{M}$ *if, for each vertex x in V, there is a d-shortest path π from x to $\mathcal{M}$ in G such that π is also a d-shortest path from x to $\mathcal{M}$ in G'.*

Let x and y be two vertices in V and let $\pi = (z_1, \ldots, z_\ell)$ be a path from x to y. We call the sum $\sum_{i=1}^{\ell-1} w(\{z_i, z_{i+1}\})$ by the *weight of* π.

Property 16. *Let d be a distance on V. Let σ_s be the operator that maps any marker set $\mathcal{M}$ into the set of partitions induced by each of the d-shortest path forests of (G, w) for $\mathcal{M}$. The operator σ_s is not hierarchical.*

Proof. Let (G, w) be the graph of Figure 2.9(a) and let d be a distance on $V(G)$ such that, for any two vertices x and y in $V(G)$, we have $d(x, y)$ equal to the minimum among the weights of all paths from x to y. Let $\mathcal{M} = \{\{a\}, \{c\}, \{f\}\}$ and $\mathcal{M}' = \{\{a\}, \{f\}\}$ be two marker sets of (G, w). In Figures 2.9(a) and (b), we show the unique d-shortest path forests of (G, w) for $\mathcal{M}$ and $\mathcal{M}'$, respectively. Hence, we have $\sigma_s(\mathcal{M}) =$

$\{\{\{a, b\}, \{c, d\}, \{e, f\}\}\}$ and $\sigma_s(\mathcal{M}') = \{\{\{a, b, c\}, \{d, e, f\}\}\}$. We can observe that the unique partition in $\sigma_s(\mathcal{M})$ is not a refinement of the unique partition in $\sigma_s(\mathcal{M}')$. Thus, σ_s is not hierarchical.



(a) (G, w) (b) G' (c) G''

Figure 2.9: From left to right: a graph (G, w), the d-shortest path forest G' (all vertices plus the blue edges) of (G, w) for the marker set $\mathcal{M} = \{\{a\}, \{c\}, \{f\}\}$, and the d-shortest path forest G'' of (G, w) for the marker set $\mathcal{M} = \{\{a\}, \{f\}\}$. The partition induced by G' is not a refinement of the partition induced by G''.

□

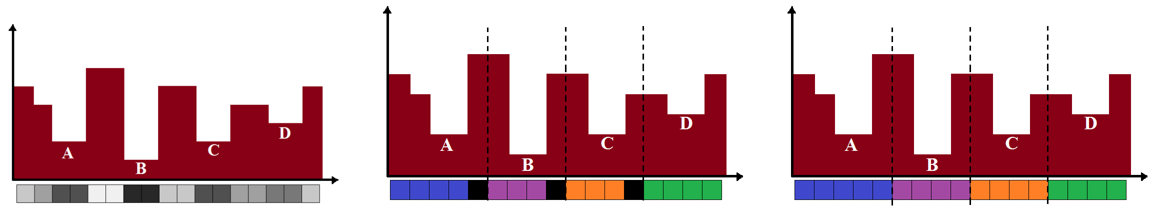
Hence, we can conclude that, unlike the MSF operator, the operators that produce min-cuts, average-cuts and shortest path forest cuts are not hierarchical.

2.6.3 Watersheds in the context of graphs and image segmentation

The watershed transform [12, 14, 22] derives from the topographic definition of watersheds lines and catchment basins. A catchment basin is a region whose collected precipitation drains to the same body of water, as a sea, and the watershed lines are the separating lines between neighbouring catchment basins. This general notion gives room to several formal definitions and algorithms to compute the watershed segmentation.

Let I be a gray-scale image. To obtain a watershed segmentation from I, we visualize I as a topographic surface in which the altitudes are given by the pixel gray-levels. A (regional) minimum of I is a set of connected pixels of uniform gray level surrounded by pixels of strictly larger gray levels. Each minimum of I is the lowest point of a catchment basins of the surface representing I. Hence, each mimimum of I induces a region in the watershed segmentation of I.

In the definition of the watershed transform, the first aspect to be considered is the definition of the watershed lines, or watershed cuts, *i.e.*, the frontiers between distinct catchment basins: either the watershed lines are a set of connected pixels separating



(a) I (b) Two watershed segmentations of I

Figure 2.10: (a) A gray-scale one-dimensional image I composed of seventeen pixels. (b) Two possible watershed segmentations of I. In the segmentation on the left, the watershed-cuts is the set of black pixels separating the catchment basins of I. In the segmentation on the right, every pixel belongs to a catchment basin of I.

distinct catchment basins, or they are the implicit frontier between distinct catchment basins (in the case where every pixel belongs to one catchment basin).

For instance, in Figure 2.10, we illustrate a gray-scale one-dimensional image I with four minima and the separation of the four catchment basins of I by two watershed segmentations.

In terms of algorithm, the watershed transform can be classified into two groups [15]: flooding based algorithms and direct search of watershed lines. The first group comprises algorithms that simulate the flooding of a topographic surface: catchment basins are iteratively immersed on water and the watershed lines are found when distinct catchment basins merge. In the algorithms of the second group, watershed lines are found by iteratively discarding pixels according to their local configuration. Furthermore, each group can be subdivided into parallel, sequential and ordered algorithms. Parallel algorithms process pixels independently: the processing of a pixel does not affect other pixels. In turn, the output of sequential algorithms entirely depends on the sequence in which pixels are processed. Ordered algorithms are also in the class of sequential algorithms except that each pixel/point is processed only once: the value of a pixel is determined after its neighbours are sufficiently known.

In [101], the authors propose an efficient linear time watershed segmentation algorithm based on recursive flooding. By simulating a flooding caused by piercing the minima of the original image and immersing this surface in water, the minima and the influence zone (catchment basin) associated to each minimum are iteratively discovered. For each k in the range of gray values of an image, the algorithm thresholds the image pixels at level k, expands the catchment basins of the already discovered minima, and find new minima of altitude k. Their algorithm assigns different labels to pixels belonging to different catchment basins, and a different label to watershed pixels. However, the watershed pixels returned by this algorithm do not necessarily separate the distinct catchment basins. Their method can be extended to general graphs and higher dimensional images.

As discussed in Section 2.6.2, watershed segmentations fall into the category of marked-based segmentations. Given a set of markers, *i.e.* subsets of image pixels or subsets of vertices, each region of a seeded segmentation includes exactly one seed. By using a watershed segmentation algorithm (*e.g.* [101]) directly on the image gradient, we implicitly define the markers as the regional minima of the gradient. Alternatively, as discussed in [75], the markers can be arbitrarily defined and the watershed segmentation algorithm is applied to a transformed version of the original gradient, in which the markers are the new regional minima.

In [2], the authors propose a stochastic approach to the watershed transform based on randomly generated markers. Given an image gradient, the authors compute several realizations of uniformly distributed sets of markers of this gradient. Then, the watershed segmentation associated to each of those realizations are combined into a new gradient, which conveys the probability of a catchment basin to be a region in the final segmentation. This new gradient can also be combined to the original gradient. The resulting segmentation is obtained by a filtered watershed segmentation of the new gradient. In the task of coarse segmentation of complex images, the authors report improved results when compared to other classical approaches.

Combined with other morphological operators and machine learning techniques, watershed segmentation is an important step for solving practical problems in many application domains such as medicine and biology, computer vision, remote sensing and material science (see *e.g.* [61, 62, 37, 44, 4, 33, 58]).

In [22], the authors formalize watersheds in the framework of edge-weighted graphs and provide the association between watershed-cuts and the MST optimization problem. They define a steepest path as a sequence of vertices such that each vertex is followed by a neighbour vertex linked by the edge of lowest weight. The catchment basin of a vertex is defined through the steepest path from this vertex to a minimum of a graph, and the watershed-cut edges are the edges linking vertices that belong to distinct catchment basins, as discussed in Section 2.6.1.

Meyer [68] generalizes the notions of watershed-cuts and steepest paths to node (vertex) and edge weighted graphs. He provides an dilation (resp. erosion) operator that, given node (resp. edge) weights, outputs edge (resp. node) weights. Meyer shows that the opening and closing resulting from the proposed dilation and erosion operators do not preserve the minima and catchment basins of a graph, which gives rise to the notion of a flooding graph: a node and edge weighted graph that is invariant under opening and closing. Then, Meyer proposes two pre-processing steps to convert any graph into a flooding graph without changing its original minima: (1) removal of every edge $u = \{x, y\}$ such that u does not belong to any steepest path, *i.e.*, u is not the edge of minimal weight among the edges adjacent to either x or y; and (2) insertion of loop edges $\{x, x\}$ such that x is the unique vertex in a minimum of the original graph. Hence, the notion of flooding graphs, which unifies the notion of watersheds in edge and node weighted graphs, allows algorithms designed for node (resp. edge) weighted graphs to be applied on edge (resp. node) weighted graphs.

As aforementioned, each minimum of a gradient induces a region in the watershed segmentation of this gradient. As a gradient usually have several non-significant minima placed in visually homogeneous regions, the watershed transform computed from the minima of a gradient is often oversegmented. As a solution to the oversegmentation resulting from the watershed transform, [15] propose the waterfall algorithm, which is an hierarchical approach to the watershed transform. At each step of the waterfall algorithm, sets of similar neighbouring catchment basins are merged, resulting in a sequence of nested segmentations.

More than dealing with the oversegmentation of the watershed transform, watershed based hierarchies find their applications in computation of morphological operators [61], iterative image segmentation [62, 37], highlighting of objects of interest [33] and hyper-spectral image analysis in the context of stochastic hierarchical watersheds [4].

In this research, we work with watershed based hierarchies of partitions that are built in the framework of weighted graphs and that are optimal in the sense of minimum spanning forests [22, 24, 27, 74]. Following the definition of [24, 27, 74], a hierarchical watershed (see Definition 1) is a sequence of partitions corresponding to filterings of an initial watershed segmentation. This sequence of filterings is guided by a total order on the minima of the original gradient, hence a total order on its catchment basins. A hierarchy following this definition is a sequence of partitions that are each optimal in the sense of minimum spanning forests: given a weighted graph (G, w) and given a hierarchical watershed $\mathcal{H}$ for a sequence $\mathcal{S} = (M_1, \ldots, M_n)$ of minima of w, each level i of $\mathcal{H}$, for i in $\{1, \ldots, n\}$, is obtained by minimizing the sum of the edges of a minimum spanning forest rooted in $\{M_i, \ldots, M_n\}$.

The link between hierarchical watersheds and the well-known minimum spanning forest optimization problem allows us to consider the results of the studies on the latter in the context of hierarchical watersheds. A first important consequence of this link is the use of minimum spanning tree algorithms to design efficient algorithms to compute hierarchical watersheds [24, 27, 74]. Moreover, the properties of minimum spanning trees induce corresponding properties on hierarchical watersheds. Furthermore, minimum spanning trees and watersheds are linked to a broader range of optimization problems. For instance, in [21], the authors unify graph-cuts, shortest path forests, watersheds and random walkers in a single framework that solves each of those problems when different parameters are given.

Most hierarchies of partitions used in the context of image analysis (*e.g.* [87]) are defined by means of an algorithm rather than by the optimization of a cost function. In turn, a hierarchical watershed optimizes a well-defined cost function for every partition. It is noteworthy that many cost functions used to define image partitions are not adapted to the computation of hierarchies. For instance, the partitions induced by the min-cuts, average-cuts and shortest path forests of a graph for a sequence of decreasing subsets of markers are generally not nested, as discussed in Section 2.6.2.

Hierarchical watersheds are also related to clustering methods oriented to energy optimization. For example, let us consider Ward's hierarchical clustering method proposed in [103]. Using this method, each level of a hierarchical cluster is optimized with respect to a cost function by evaluating all possible $\frac{n\times(n-1)}{2}$ merging of regions (considering that the current level has n regions). Ward's method is greedy because the optimal mergings are selected independently at each level. However, Ward's cost function needs to be updated after each merging, which leads to a cubical time complexity algorithm. In turn, considering the objective function optimized by hierarchical watersheds, we are able to greedily find the optimal pair of regions to be merged without having to recompute the cost function.

In [31], the authors propose the Image Foresting Transform (IFT) as a global framework to solve graph-cut problems such as the watershed transform and other marker-based segmentations. Their method can be applied to a family of *smooth* energy functions which can be optimized locally. Definitions of the watershed segmentation, including the watershed segmentation resulting from the IFT, are discussed in [10].

2.7 Attribute based hierarchies

An attribute based hierarchy is a hierarchy of partitions whose construction is guided by a criterion A (or several criteria $A_1, A_2, \dots$) such that, at the highest levels of the hierarchy, the most important regions according to A are preserved. In this section, we first present a brief review on attribute morphological operators and their applications. Then, we introduce the increasing criteria used in the computation of hierarchical watersheds. Finally, we discuss hierarchies based on non-increasing criteria.

2.7.1 Introduction

As discussed in Section 2.2, the optimal hierarchy of segmentations of an image is application dependent. The criterion upon which the relevant regions of an image can be described varies according to the application, *e.g.* the segmentation of circular regions used in the search of traffic signs in [104] is not adapted to the detection of (rectangular) documents discussed in the same article. Among the most commonly used handcrafted criteria, we can cite area/surface, contrast and shape based criteria, *e.g.* circularity, rectangularity and perimeter.

Let $\mathcal{H}$ be a hierarchy and let A be a criterion on the regions of $\mathcal{H}$. For instance, if A is the area criterion, then for any region R of $\mathcal{H}$, we denote by $A(R)$ the area (number of pixels) of R. We say that A *is increasing* if, given two regions R_1 and R_2 of $\mathcal{H}$ such that $R_1 \subset R_2$, we have $A(R_1) \leq A(R_2)$. Otherwise, we say that A *is non-increasing.* For instance, the area attribute is increasing: the number of pixels of a region is always greater than the number of pixels of its children. On the other hand, most shape based criteria are non-increasing: a non-circular region can be the parent of a circular region. Those two type of criteria lead to distinct attribute based hierarchies.

Attribute based hierarchies are intrinsically related to attribute morphological operators such as attribute opening/closing. When dealing with an increasing criterion A, a hierarchy of segmentations can be designed as a sequence of "A openings" (of a given initial segmentation) with increasing thresholds values. In order words, given an increasing attribute A, we expect the regions of the highest levels of the hierarchy based on A to have the highest values for A. In [18], the authors discuss attribute morphological operators and the difficulty of handling non-increasing criteria in morphological operators which were designed to be extensive or anti-extensive.

Let us consider a hierarchy $\mathcal{H}$ and an increasing criterion A. The filtering of $\mathcal{H}$ with respect to A and a parameter λ corresponds to the pruning of every region R of $\mathcal{H}$ such that $A(R) < \lambda$. When handling increasing attributes, the removal of a node in a hierarchy implies the removal of all the descendants of this node. However, this is not the case for non-increasing attributes. In [88], the authors propose three approaches to handle prunings based on non-increasing criteria:

Direct. a node R is removed if $A(R) < \lambda$.

Max. a node R is removed if $A(R) < \lambda$ and if, for any region R' such that $R' \subset R$, we also have $A(R') < \lambda$.

Min. a node is removed if $A(R) < \lambda$ or if there is a region R' such that $R \subset R'$ and $A(R') < \lambda$.

Viterbi. a cost is assigned for the removal or preservation of each region of the hierarchy. The optimal solution is found by the optimal path from each leaf region to the root of the hierarchy, where each node is associated to two possible states: remove or preserve. This technique shows improved results when compared to the other three techniques in the sense that similar images get similar filtering results [88].

Until now, we have discussed attribute filters in which all regions of a hierarchy are considered during the filtering. Alternatively, filters can be extrema oriented in the sense that they act on the extrema of an image: each extremum (maximum or minimum) is either preserved or completely removed. In [98], the authors introduce the notion of extinction values associated to the extrema of an image. Given the component tree T of an image I, the extinction value of an extremum M of T with respect to an attribute A corresponds the persistence of M when a A-filter is applied to T: the extinction value of M is the largest value λ such that M belongs to an extremum of the A-filter of T with parameter λ. Extrema oriented operators guided by extinction values are called extinction operators.

In [93], the authors state the superiority of extinction filters with respect to attribute filters in image simplification applied to recognition tasks. The former depends on threshold parameters while that the latter is guided by the number of extrema we aim to preserve. Being so, extinction filters have the advantage of being less scale dependant when compared to attribute filters [41].

In [98], extinction values are defined for the attributes area and volume. Then, this notion was extended to other increasing topological criteria such as height, number of descendants and diagonal of bounding box proposed in [91], and number of parent nodes introduced in [79]. Extinction values are also related to the notion of dynamics of minima introduced in [45], the dynamics of a minimum being a measure of contrast between the minima of an image.

In [91], the authors present applications of their newly proposed extinction values in the detection of objects: they use the max-tree or min-tree of an image to detect the extrema of interest in the image, where the extrema are normally associated to relevant objects.

In [106], the authors propose a method to extend the notion of extinction values to non-increasing criteria. They propose the computation of the max-tree from the original max-tree. The attribute values on this new max-tree is increasing.

In [41], the authors present a fully-automatic solution to the segmentation of remote sensing data based on extinction filters. They use extinction values based on the increasing criteria area, volume, height and diagonal of bounding box, and on the non-increasing standard deviation criterion. They handle non-increasing attributes as proposed in [106] and, then, they apply their method on different remote sensing datasets for a classification purpose.

In [56], the authors define a super-pixel method based on watersheds, called water-pixels, with spatially uniformly distributed markers. At each cell in an spatially uniform grid (rectangular or hexagonal), one of the minima included in the cell is chosen according to their extinction values: the minima of greatest area extinction value is preserved. Then, they regularize the image gradient according to the chosen markers.

In [40], the authors apply the notions of extinction values and convolutional neural networks to fuse the complementary features of hyperspectral and LiDAR data.

Attribute based hierarchies can be computed from different methods. In [104], the authors proposed hierarchies based on non-increasing attributes computed from the tree of shapes and from extinction values. In [98], the authors proposed hierarchies based on area and volume extinction values. In [33], the authors compute attribute based hierarchies by the sequential combination of stochastic watershed hierarchies based on distinct attributes.

2.7.2 Hierarchical watersheds based on extinction values

Extinction values are used by extrema-oriented connected filters. When filtering by extinction values, we are interested in either keeping or completely removing a maxima or minima of an image. This notion fits the definition of hierarchical watersheds in the sense that, at each level of a hierarchical watershed, we are interested in filtering out one catchment basin. The bijection between the minima and the catchment basins of an image make extinction values very adapted to the construction of hierarchical watersheds. More precisely, the sequence of minima for which a hierarchical watershed is constructed can be ranked by extinction values computed from any increasing criterion.

Given an image I and the max-tree or min-tree T of I, the extinction values of the extrema of I with respect to an increasing attribute A can be recursively computed from the root to the leaves of T as follows. The root of T is assigned to ∞. Then, for each child R of the current node, if the attribute value of R for A is greater than the attribute values of all its siblings, then R receives the extinction values of its parent. The extinction value of the other siblings is assigned to their own attribute values. When there are ties between siblings, they are treated arbitrarily.

In the following, we list the increasing attributes whose extinction values are used to compute hierarchical watersheds in this research. In the remainder of this section, we consider a connected weighted graph (G, w) and its min-tree MT. Extinction values are recursively computed from the root to the leaves of MT as described previously.

Area (surface): the area of a region R of MT is the number of pixels of R. Area extinction values were introduced in [98]. Since then, it has found its application on the segmentation of microarray images (filtering step aiming to reduce noise) [3], classification of remote sensing data [41] and computation of spatially regular super pixels (waterpixels) [56].

Dynamics: the dynamics [45] measures the importance of a minimum with respect to other neighbouring extrema. Let Min be a minimum of MT. The dynamics of Min is the lowest difference between the altitude of Min and the altitude of an edge on a path from Min to another minimum Min' of (G, w) such that the altitude of Min' is lower than the altitude of Min. In [75], the dynamics of minima is extended to the dynamics of contours, leading to nested closed contours (saliency map), which are applied to shape recognition. In [36], the authors apply the dynamics of maxima in the search for markers for motion analysis from several frames of a video.

Volume: volume extinction values, introduced in [98], combine area and contrast information. The volume of a node R of MT is the sum $\sum_{R' \subset R} Area(R') \times C(R')$ such that $C(R')$ is the altitude of R' minus the altitude of the parent of R'. In [41], volume extinction values are used in the segmentation of remote sensing data.

In the following, we present topological based criteria. As topological attributes, they are invariant to monotone contrast transformations and to geometric transformations.

Number of descendants: introduced in [91], the number of descendants of a node R of MT is the number of regions of MT that are included in R. In [91], the authors use extinction values based on the number of descendants in the detection of objects, where the objects of an image are expected to be an extremum of the original image.

Topological height: number of nodes in the longest path from a node R to a leaf node R' of MT [91].

Diagonal of bounding box: diagonal of the smallest bounding box covering a region R. The bounding box is parallel to the x and y axis of the image. It summarizes the width and height of bounding box attributes proposed in [91].

Number of minima: number of minima of w that are subsets of a region R of MT.

Number of parent nodes: a parent node of MT is a node that is the parent of at least one node of MT. The number of parent nodes of a region R is the number of parent nodes that are a subset of R (R included). We introduced this measure in [79] and, according to the evaluation framework proposed in the same article, hierarchical watersheds based on this new attribute outperform hierarchical watersheds based on the aforementioned increasing attributes (see Chapter 6).

In Figure 2.11, we illustrat hierarchical watersheds computed from the extinction values obtained from each of the increasing attributes presented in this section.

2.7.3 Hierarchies based on non-increasing attributes

Computing hierarchical watersheds from extinction values based on increasing attributes ensures that the higher levels of the hierarchy contains the most relevant regions according to a given criterion. However, this property can not be assured when non-increasing attributes are considered. This problem has been tackled in [88, 106] in the context of connected filtering and hierarchies based on shape attributes.

In this work, we consider three shape-based non-increasing criteria:

Perimeter: given a region R, the perimeter of R is the number of pixels in R that are 4-connected to the background. The pixels in the image boundaries are considered to be connected to the background. We denote the perimeter of R by $\mathcal{P}er(R)$.

Circularity: we consider a well-known circularity measure based on area and perimeter. Given a region R, let $\mathcal{A}(R)$ be the number of pixels in R. The circularity of R, denoted by $\mathcal{C}ir(R)$, is $\frac{4\times\pi\mathcal{A}(R)}{\mathcal{P}er(R)^2}$. In the continuous case, the value $\mathcal{C}ir(R)$ is in the range $[0, 1]$ with $\mathcal{C}ir(R) = 1$ when R is a circle. In the discrete case, the maximal value of $\mathcal{C}ir(R)$ is greater than one. Still, this measure is able to distinguish approximately circular regions and non-circular regions (see Figure 2.12).

Rectangularity: given a region R, the rectangularity of R is measured by the area of R divided by the area of the bounding box of R: $\frac{\mathcal{A}(R)}{\mathcal{A}_{bb}(R)}$. As the bounding box of a region is parallel to the x and y axis, this measure is sensitive to rotation. Since the area of a region is always smaller than the area of its bounding box, the rectangularity of R, denoted by $\mathcal{R}ec(R)$, is in the range $(0, 1]$. As we work with regular rectangular grids, this measure reaches 1 when R is a rectangle.

In Figure 2.12, we illustrate the three shape-based criteria considered here. We can observe that the circularity and rectangularity measures succeed at distinguishing rectangular from circular shapes.

To compute hierarchical watersheds from non-increasing attributes, we first consider extinction values computed from regularized attribute values. After regularization, the non-increasing attributes become increasing attributes such that, filtering a hierarchy based on those increasing attributes is equivalent to using the max and min rules on the non-increasing attributes. Using the max rule (resp. min rule), the attribute of a node R is assigned to the maximum (resp. minimum) of the attribute values among the descendants of R (R included).

In Figure 2.13, we show two hierarchical watersheds based on extinction values computed from regularized circularity. We can observe that neither of those hierarchical

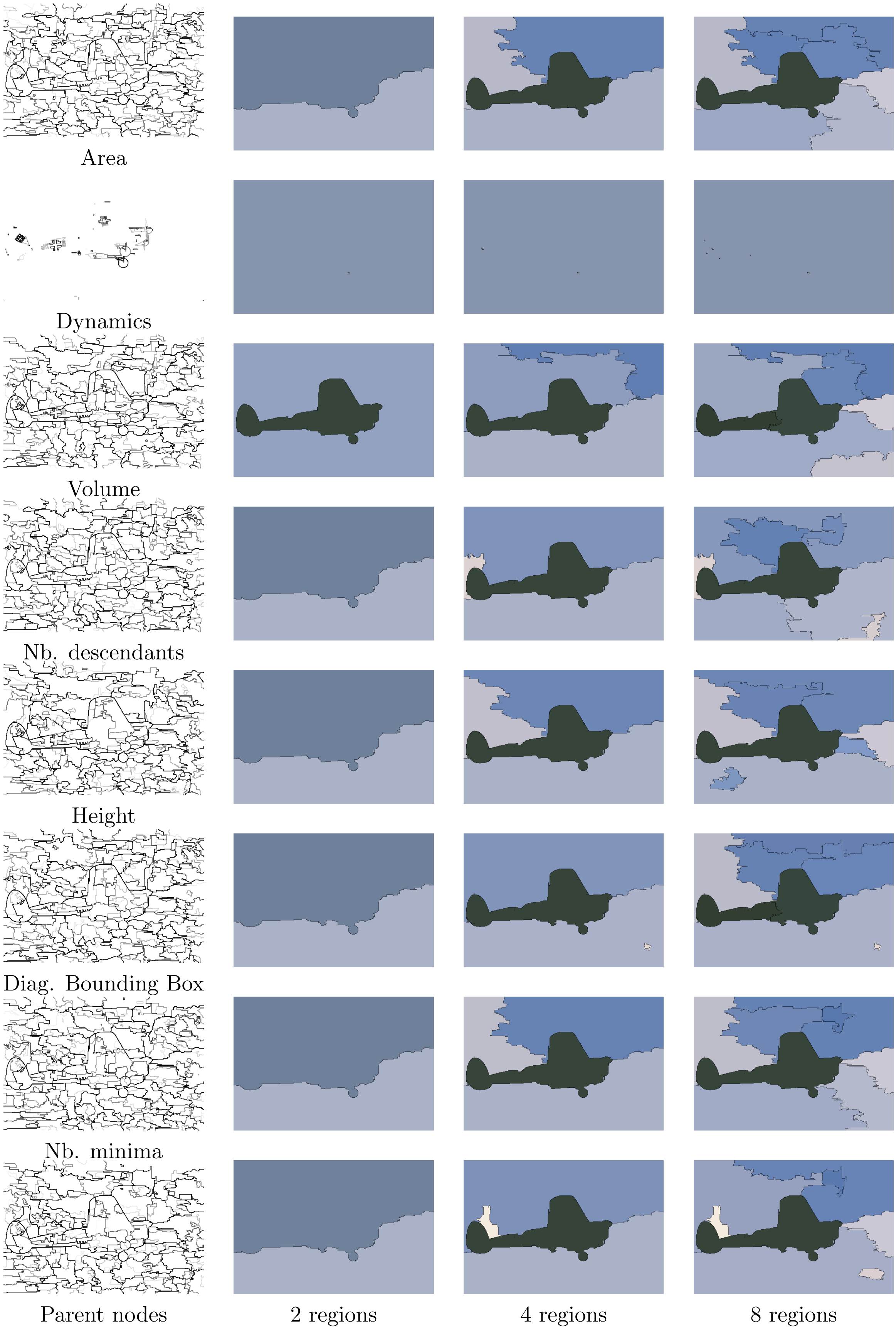


Figure 2.11: The saliency maps of the hierarchical watersheds of an image based on the extinction values of area, dynamics, volume, number of descendants, topological height, diagonal of bounding box, number of minima and number of parent nodes.

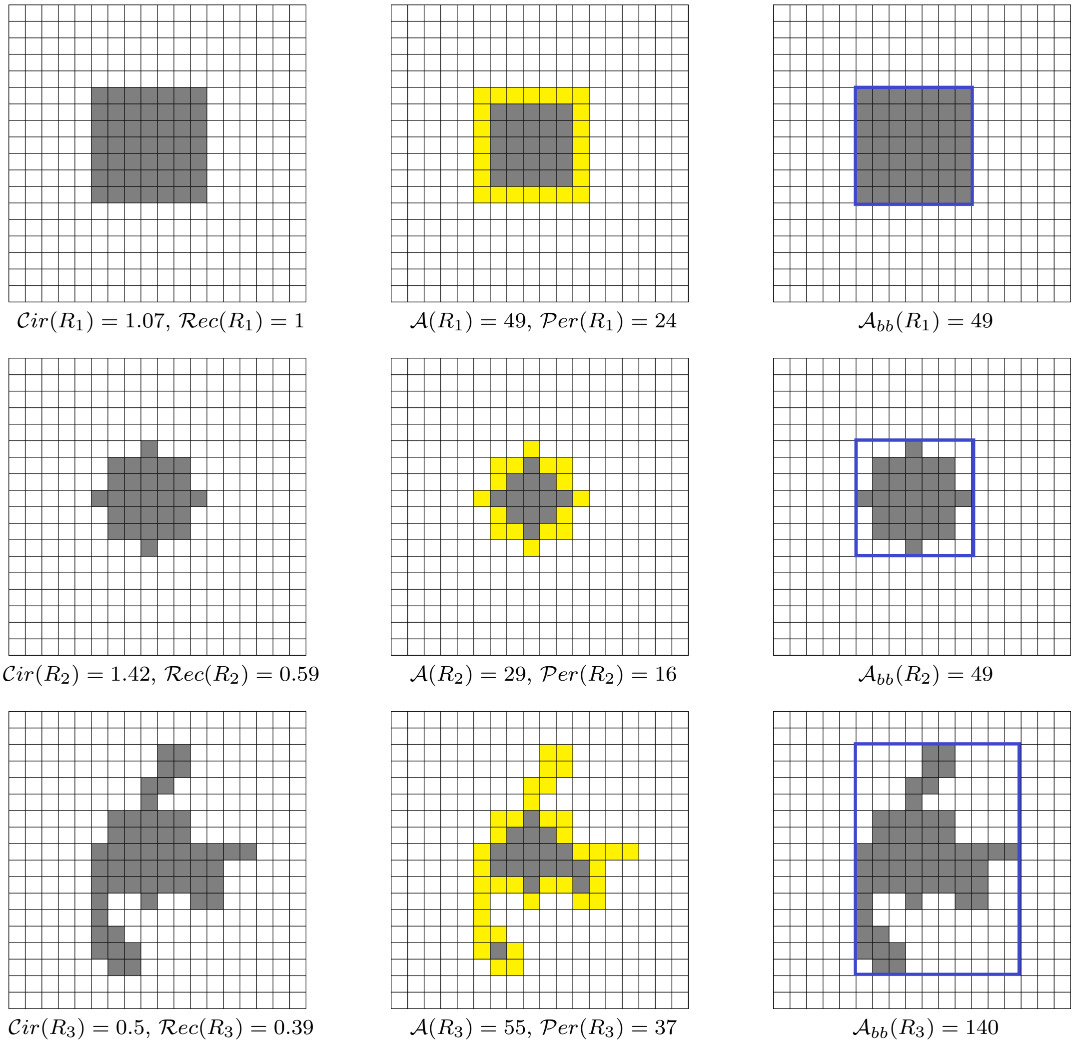


Figure 2.12: Circularity, rectangularity, area, perimeter and area of bounding box of three shapes in the rectangular grid.

watersheds succeed at highlighting the most circular regions of the original image. Using the max-rule, non-circular regions are preserved at high levels of the hierarchy simply because they include circular regions. Using the min rule, circular regions are rapidly filtered out because they are included in non-circular regions.

To take into account the cases where important regions can be characterized by non-increasing attributes, as circularity and perimeter, we apply a simple method to extract and stack those regions in a hierarchy of segmentations. Given a min-tree of an image I (or any hierarchy computed from I), we compute non-increasing attribute values on each node of the tree. Then, a new saliency map is constructed by assigning to each edge the maximum attribute value among the regions adjacent to it.

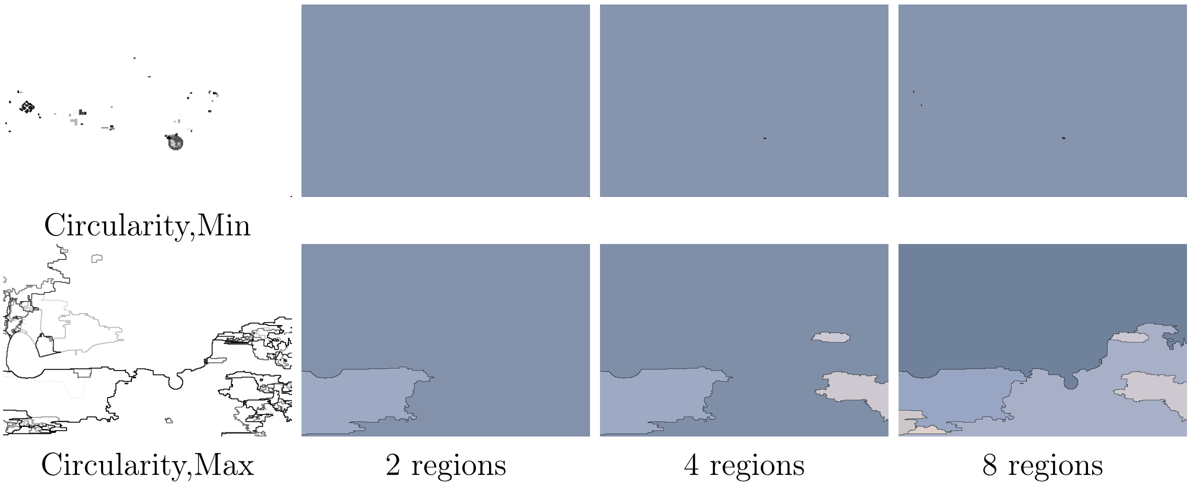


Figure 2.13: Circularity based hierarchical watersheds. The circularity attributes were normalized using max and min rules.

Examples of hierarchies based on circularity are given in Figure 2.14. From the area based hierarchical watersheds of each image of Figure 2.14, we computed the circularity measure and proceeded as described in the previous paragraph. For visualization purposes, only the highest levels of each hierarchy are shown.

A different approach to handle non-increasing attributes is proposed in [104]. In [104], the authors first compute a component tree T (min-tree, max-tree or tree of shapes) of the original image. Then, non-increasing attribute values are computed for each node of the component tree. A new component tree TT is computed from T by considering the attribute value of each node as its gray-level. We can affirm that there is a bijection between the nodes of TT and the nodes of T. The leaves of TT are the nodes of T with the lowest attribute values. Hence, the non-increasing criteria becomes increasing on the new tree TT. The filtering is then performed on this new tree TT.

It is important to note that hierarchies based on non-increasing attributes are not watershed hierarchies in general and, for this reason, none of the regularization strategies proposed in [88] would be able to produce hierarchical watersheds that highlight the desired regions at the highest levels.

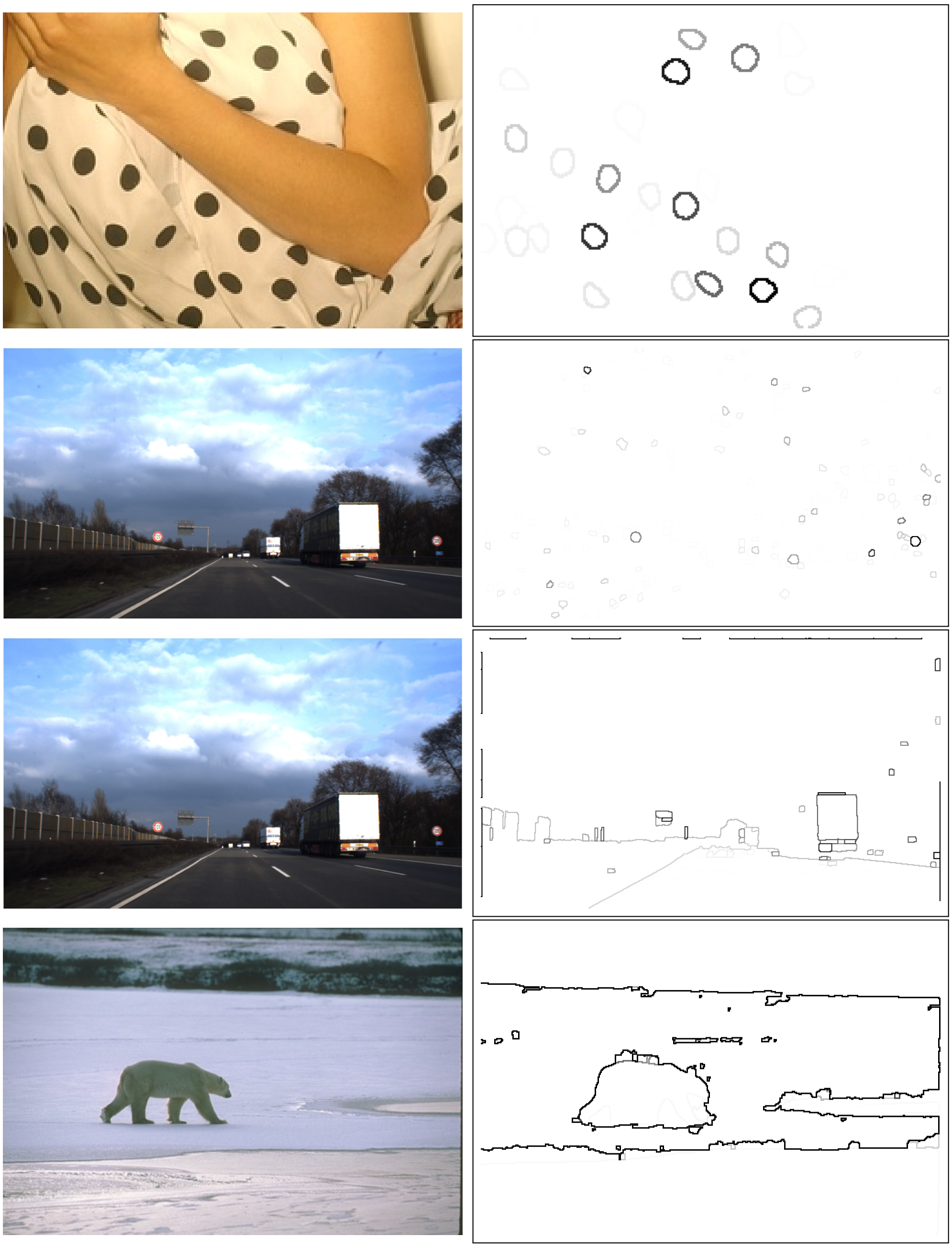


Figure 2.14: From first to forth rows: original image and the saliency maps of the hierarchies based on circularity, circularity, rectangularity and perimeter, respectively.

Chapter 3

Characterization and recognition of hierarchical watersheds

In this chapter, we propose a characterization of hierarchical watersheds and an algorithm to recognize hierarchical watersheds. The results herein presented are based on the following articles:

- D. S. Maia, J. Cousty, L. Najman, and B. Perret. Recognizing hierarchical watersheds. In *International Conference on Discrete Geometry for Computer Imagery*, pages 300–313. Springer, 2019.
- D. S. Maia, J. Cousty, L. Najman, and B. Perret. Characterization of graph based hierarchical watersheds: theory and algorithm. Under review. 2019.

3.1 Introduction

As discussed in Section 2.6, watershed [14, 22] is a well established segmentation technique in the field of mathematical morphology. The idea behind this technique is related to the topographic definition of watersheds: dividing lines between neighboring catchment basins, *i.e.*, regions whose collected water drains to a common point. We say that the point (or region) of lowest altitude of a catchment basin is a (local) minimum of a topographic surface. In the context of digital image processing, gray-level images (gradients) can be treated as topographic surfaces whose altitudes are determined by the pixel gray-levels. The minima of an image are the regions of uniform gray-level surrounded by pixels of strictly greater gray-levels. A watershed segmentation is a partition of the set of pixels of an image into its catchment basins.

Hierarchical watersheds [24, 13, 75, 67], reviewed in Section 2.6.1, are sequences of nested partitions which correspond to filterings of an initial watershed segmentation [22, 14]. Given an image I, a hierarchical watershed of I can be obtained by iteratively merging neighboring catchment basins of I according to a predefined ordering on the minima of I.

Several well-known image segmentation techniques are modeled in the framework of graphs [17, 90, 35, 31, 43, 21], including (hierarchical) watersheds [66, 22, 24, 27, 74]. In this context, images are often represented as (edge) weighted graphs whose vertices correspond to pixels and whose edge weights convey the dissimilarity between neighboring pixels. Let G be a graph whose edges are weighted by a map w. As defined in Section 2.6.1, a minimum of w is a subgraph of G with equal edge weights that is surrounded by edges with strictly greater weights. A hierarchical watershed of (G, w) for a sequence of minima $\mathcal{S}$ of w is constructed by merging the catchment basins of (G, w) following the sequence $\mathcal{S}$.

Hierarchical watersheds can feature several distinct aspects of an image. As established in Section 2.7.2, the minima of a weighted graph are commonly ordered by extinction values based on a regional attribute A, *e.g.* area and volume [98]. We then expect the resulting hierarchical watershed to highlight the most perceptually significant regions with respect to this attribute A. Besides being versatile, hierarchical watersheds can be computed by the efficient algorithm proposed in [27, 74], whose time complexity is the same as minimum spanning tree algorithms. Moreover, as shown in [79], the performance of hierarchical watersheds based on regional attributes is competitive when compared to other hierarchical segmentation methods.

The link between hierarchical watersheds and the well-known minimum spanning forest optimization problem, discussed in Section 2.6, allows us to consider the results of the studies on the latter in the context of hierarchical watersheds. A first important consequence of this link is the use of minimum spanning tree algorithms to design efficient algorithms to compute hierarchical watersheds [24, 27, 74]. Moreover, the properties of minimum spanning trees induce corresponding properties on hierarchical watersheds. Furthermore, minimum spanning trees and watersheds are linked to a broader range of optimization problems. For instance, in [21], the authors unify graph-cuts, shortest path forests, watersheds and random walkers in a single framework that solves each of those problems when different parameters are given.

Most hierarchies of partitions used in the context of image analysis (*e.g.* [87]) are defined by means of an algorithm rather than by the optimization of a cost function. In turn, a hierarchical watershed optimizes a well-defined cost function for every partition.

It is noteworthy that many cost functions used to define image partitions are not adapted to the computation of hierarchies. For instance, the partitions induced by the min-cuts, average-cuts and shortest path forests of a graph for a sequence of decreasing subsets of markers are generally not nested (see examples in Section 2.6.2).

In this study, we tackle the problem of recognizing hierarchical watersheds:

- given a weighted graph (G, w) and a hierarchy of partitions $\mathcal{H}$, determine if $\mathcal{H}$ is a hierarchical watershed of (G, w).

The problem of recognizing hierarchical watersheds is related to the problem studied in [55, 8]. In [55, 8], the authors search for a minimum set of markers which lead to a given watershed segmentation. In our case, we first consider a fixed set of markers, namely the set of minima of a graph (G, w). Then, given a hierarchy, we investigate if, for every partition $\mathbf{P}$ of this hierarchy, there is a subset $\mathcal{M}$ of markers (minima of (G, w)) such that $\mathbf{P}$ is the watershed segmentation for $\mathcal{M}$, *i.e.*, the connected components of a MSF rooted in $\mathcal{M}$ (see the definition of hierarchical watersheds in Definition 1).

A naive approach to solve this problem is to test if there is a sequence $\mathcal{S}$ of minima of w such that $\mathcal{H}$ is the hierarchical watershed of (G, w) for $\mathcal{S}$. However, there exist $n!$ sequences of minima of w, which leads to an algorithm of factorial time complexity.

Motivated by solving this recognition problem more efficiently, we propose in Section 3.2 a simple and general characterization of hierarchical watersheds (Theorem 20) based on the binary partition hierarchy by altitude ordering, whose definition was given in Section 2.5.2. Based on our proposed characterization of hierarchical watersheds, we design an efficient algorithm (Algorithm 1) to solve this problem. Then, in Section 3.4, we introduce a relaxed definition of hierarchical watersheds, called flattened hierarchical watersheds, along with an algorithm to recognize this family of hierarchies.

Important notations: in the remainder of this chapter, the symbol (G, w) denotes a weighted graph whose vertex set is connected. To shorten the notation, the vertex set of G is denoted by V and its edge set is denoted by E. Without loss of generality, we also assume that the range of w is included in the set $\mathbb{E}$ of all integers from 0 to $|E| - 1$ (otherwise, one could always consider an increasing one-to-one correspondence from the set $\{w(u) \mid u \in E\}$ into the subset $\{0, ..., |\{w(u) \mid u \in E\}| - 1\}$ of $\mathbb{E}$). Every hierarchy considered in this chapter is connected for G and therefore, for the sake of simplicity, we use the term *hierarchy* instead of *hierarchy which is connected for G*. We denote by n the number of minima of w. Every sequence of minima of w considered in this chapter is a sequence of n pairwise distinct minima of w and, therefore, for the sake of simplicity, we use the term *sequence of minima of w* instead of *sequence of n pairwise distinct minima* *of* w. By abuse of terminology, when no confusion is possible, if M is a minimum of w, we call the set $V(M)$ of vertices of M as a minimum of w.

3.2 Characterization of hierarchical watersheds

In this section, we propose a characterization of hierarchical watersheds for solving the problem of recognizing hierarchical watersheds in the context of weighted graphs. This characterization relies on the bijection between hierarchies of partitions and saliency maps. As discussed in Section 2.4, any hierarchy of partitions has a unique saliency map. In turn, a map can be the saliency map of at most one hierarchy of partitions. Therefore, by characterizing the saliency maps of hierarchical watersheds, we characterize hierarchical watersheds as well. To ease the reading of this section, the proofs of the properties and theorems stated here are delayed to Appendix 8.1.

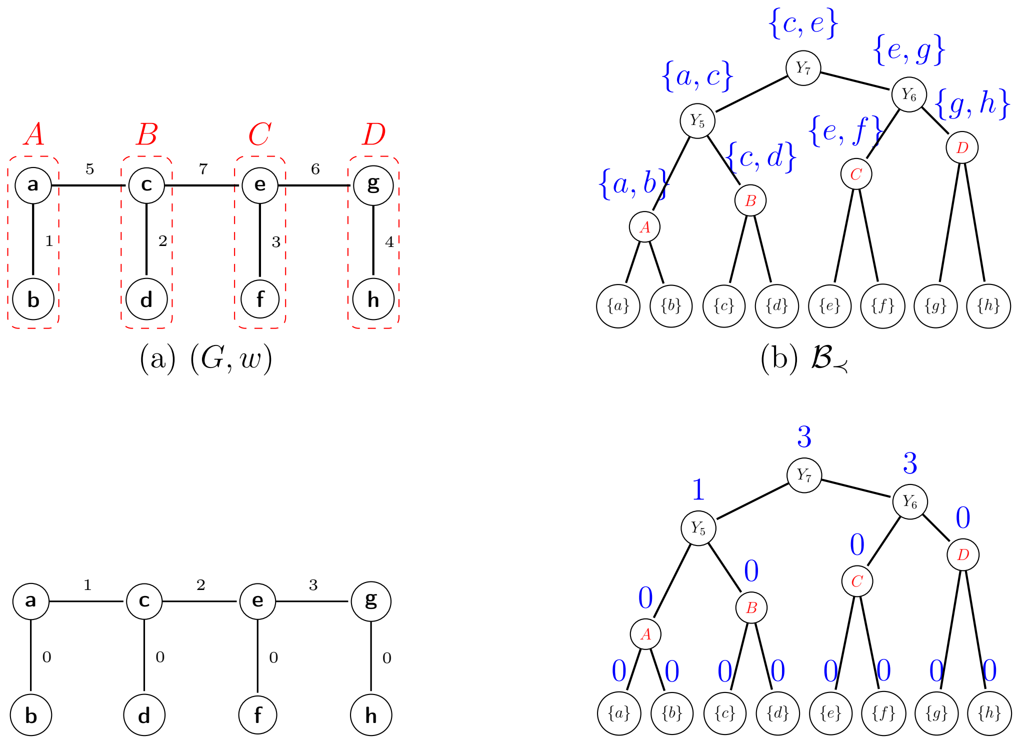
The characterization of (the saliency map of) hierarchical watersheds presented in this chapter is based on the link between hierarchical watersheds and binary partition hierarchies (by altitude ordering) presented in Section 2.5.2.

Important notations: given an altitude ordering $\prec$ for w and a building edge u for $\prec$ (see definitions in Section 2.5.2), we denote by R_u the region of the binary partition hierarchy $\mathcal{B}_\prec$ by $\prec$ whose building edge is u. The set of building edges for $\prec$ is denoted by $E_\prec$.

Definition 18, presented in the following, introduces the notion of one-side increasing map. As established later in Lemma 19, this notion is closed linked to the saliency maps of hierarchical watersheds. Before introducing one-side increasing maps, we present the auxiliary notion of supremum descendant value in the next definition.

Definition 17 (supremum descendant value)**.** *Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$. Let R be a region of the binary partition hierarchy $\mathcal{B}_\prec$ of (G, w) by $\prec$. The* supremum descendant value of R for $(f, \prec)$ *is the supremum edge weight among the building edges of the regions included in R: $\vee\{f(v) \mid v \in E_\prec, R_v \subseteq R\}$, where $\vee\{\} = 0$.*

For instance, let (G, w) be the graph of Figure 3.1(a). Since the edges of G have pairwise distinct weights in w, we can conclude that there is a unique altitude ordering for w. Let $\prec$ be the altitude ordering for w and let $\mathcal{B}_\prec$ be the binary partition hierarchy by $\prec$ of Figure 3.1(b). Given the map f illustrated in Figure 3.1(c), we can verify that the weights above the nodes of $\mathcal{B}_\prec$, illustrated in Figure 3.1(d), are the supremum descendant values for $(f, \prec)$.



(c) (G, f) (d) supremum descendant values for $(f, \prec)$

Figure 3.1: (a) a weighted graph (G, w) with four minima delimited by the dashed rectangles. (b): the binary partition hierarchy $\mathcal{B}_\prec$ by the unique altitude ordering $\prec$ for w. The building edges for $\prec$ are shown above each non-leaf region of $\mathcal{B}_\prec$ (c) a map f. (d) the supremum descendant values for $(f, \prec)$.

Definition 18 (one-side increasing map)**.** *Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$. We say that f is* one-side increasing for $\prec$ *if:*

1. $\{f(u) \mid u \in E_\prec\} = \{0, \ldots, n-1\}$*;*

2. *for any edge u in $E_\prec$, the value $f(u)$ is greater than zero if and only if u is a watershed-cut edge for $\prec$ (Definition 2); and*

3. *for any edge u in $E_\prec$, there exists a child R of R_u such that $f(u)$ is greater than or equal to the supremum descendant value of R for $(f, \prec)$.*

The next lemma, whose proof is given in Appendix 8.1.1, states that hierarchical watersheds can be characterized as the hierarchies whose saliency maps are one-side increasing maps.

Lemma 19. *Let $\mathcal{H}$ be a hierarchy on V. The hierarchy $\mathcal{H}$ is a hierarchical watershed of (G, w) if and only if there is an altitude ordering $\prec$ for w such that the saliency map $\Phi(\mathcal{H})$ is one-side increasing for $\prec$.*

Let (G, w) be the graph of Figure 3.2(a), let $\prec$ be the unique altitude ordering for w and let $\mathcal{B}_\prec$ be the binary partition hierarchy by $\prec$ shown in 3.2(b). Let $\mathcal{H}$ be the hierarchy of Figure 3.2(c) and let $\Phi(\mathcal{H})$ be the saliency map of $\mathcal{H}$ shown in Figure 3.2(d). In Figure 3.2(e), the saliency map $\Phi(\mathcal{H})$ is represented on the hierarchy $\mathcal{B}_\prec$. We can verify that $\Phi(\mathcal{H})$ is one-side increasing for $\prec$. By Lemma 19, we may affirm that $\Phi(\mathcal{H})$ is the saliency map of a hierarchical watershed of (G, w) and that, consequently, the hierarchy $\mathcal{H}$ is a hierarchical watershed of (G, w). Indeed, we can verify that $\mathcal{H}$ is the hierarchical watershed for the sequence (A, B, C, D) of minima of w.

Let $\mathcal{H}'$ and $\Phi(\mathcal{H}')$ be the hierarchy and the saliency map shown in Figure 3.2(f) and Figure 3.2(g), respectively. We can see that $\Phi(\mathcal{H}')$ is not one-side increasing for $\prec$. Indeed, the weight $\Phi(\mathcal{H}')(\{c, e\})$ of the building edge of the region Y_7 of $\mathcal{B}_\prec$ is 1, which is lower than both $\vee\{\Phi(\mathcal{H}')(v) \mid R_v \subseteq Y_5\} = 2$ and $\vee\{\Phi(\mathcal{H}')(v) \mid R_v \subseteq Y_6\} = 3$. Hence, the condition 3 of Definition 18 is not satisfied by $\Phi(\mathcal{H}')$. Thus, by Lemma 19, as $\prec$ is the unique altitude ordering for w, we deduce that $\Phi(\mathcal{H}')$ is not the saliency map of a hierarchical watershed of (G, w) and that $\mathcal{H}'$ is not a hierarchical watershed of (G, w). Indeed, it can be verified that there is no sequence $\mathcal{S}$ of minima of w such that $\mathcal{H}'$ is the hierarchical watershed of (G, w) for $\mathcal{S}$.

In the case where (G, w) has pairwise distinct edge weights, there exists a unique altitude ordering for w. Hence, we can use Lemma 19 to verify that a given map f is the saliency map of a hierarchical watershed of (G, w) by simply checking if f is one-side increasing for the unique altitude ordering for w. Otherwise, let us consider that (G, w) has arbitrary edge weights. Thus, in order to test if a map f is the saliency map of a hierarchical watershed of (G, w), we need to test if there is an altitude ordering $\prec$ for w such that f is one-side increasing for $\prec$. In the worst case, there exist $|E|!$ possible altitude orderings for w. Hence, the naive approach to verify that f is one-side increasing for an altitude ordering for w has a factorial time complexity, which is the same time complexity as the algorithm to verify that f is the saliency map of a hierarchical watershed for a sequence of minima of w. Actually, as we establish later in Theorem 20, it is sufficient to test if f is one-side increasing for a single altitude ordering for w, which is the key idea behind our efficient algorithm (Algorithm 1) to recognize hierarchical watersheds.

Let f and g be two maps from E into $\mathbb{R}$. A *lexicographic ordering for* (f, g) is a total ordering $\prec$ on E such that, for any two edges u and v in E, we have $u \prec v$ if $f(u) < f(v)$ or if $f(u) = f(v)$ and $g(u) \leq g(v)$. We can note that any lexicographic ordering for (f, g)

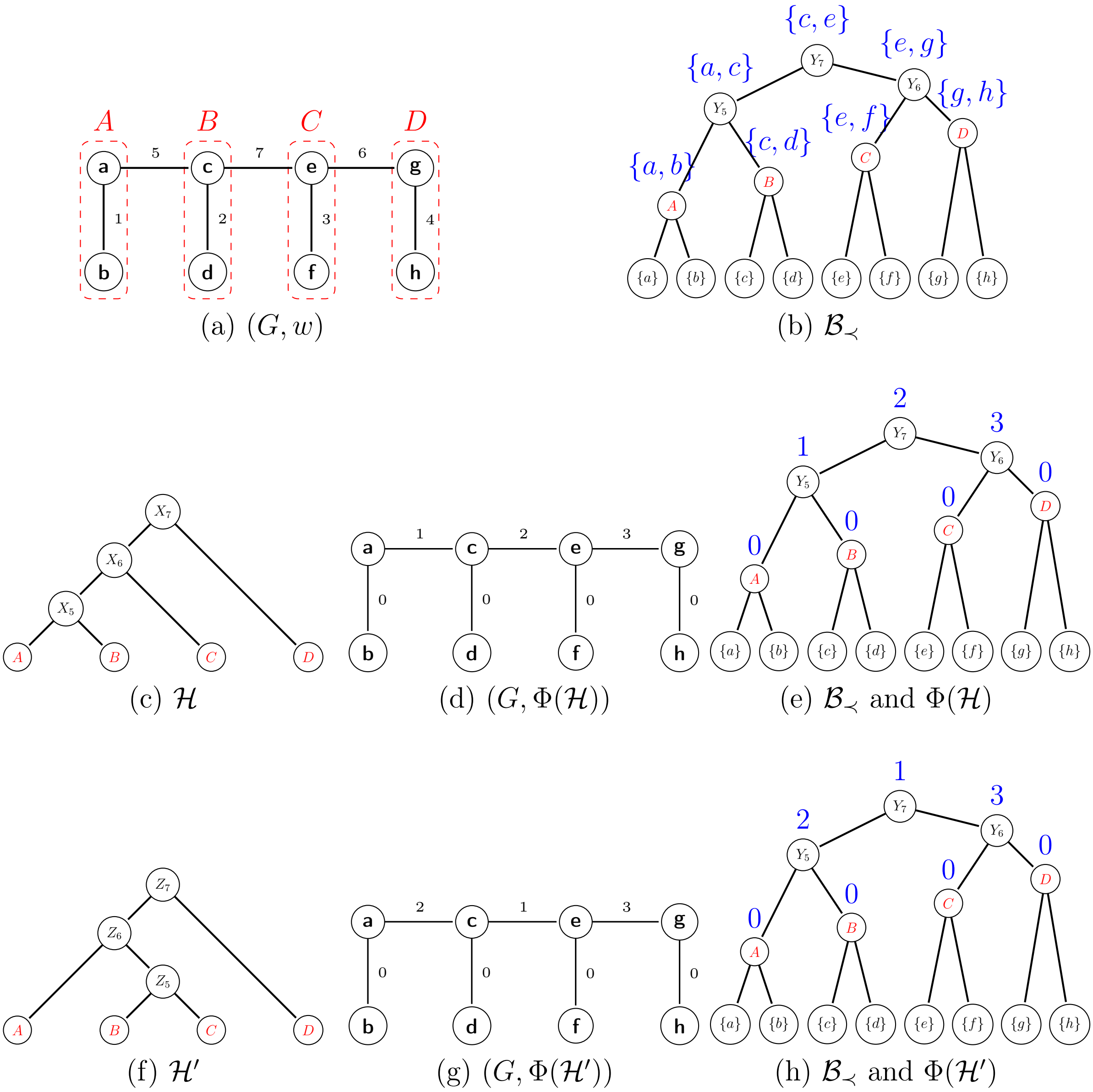


Figure 3.2: (a) a weighted graph (G, w) with four minima limited by the dashed rectangles. (b): the binary partition hierarchy $\mathcal{B}_{\prec}$ by the unique altitude ordering $\prec$ for w. (c) and (f): the hierarchies $\mathcal{H}$ and $\mathcal{H}'$, respectively. (d) and (g): the saliency maps of $\mathcal{H}$ and $\mathcal{H}'$, respectively. (e) and (h): the maps $\Phi(\mathcal{H})$ and $\Phi(\mathcal{H}')$ represented on the hierarchy $\mathcal{B}_{\prec}$. For each edge u of G, the values $\Phi(\mathcal{H})(u)$ and $\Phi(\mathcal{H}')(u)$ are shown above the region R_u of $\mathcal{B}_{\prec}$.

is an altitude ordering for f.

Theorem 20. *Let $\mathcal{H}$ be a hierarchy and let $\prec$ be a lexicographic ordering for $(w, \Phi(\mathcal{H}))$. The hierarchy $\mathcal{H}$ is a hierarchical watershed of (G, w) if and only if the saliency map $\Phi(\mathcal{H})$ is one-side increasing for $\prec$.*

The proof of Theorem 20 is presented in Appendix 8.1.2.

3.3 Algorithm to recognize hierarchical watersheds

In this section, we present an efficient algorithm to recognize hierarchical watersheds based on Theorem 20. To test if a hierarchy is a hierarchical watershed of (G, w), it is sufficient to verify that the saliency map f of this hierarchy is one-side increasing for a lexicographic ordering for (w, f).

Algorithm 1 provides a description of our algorithm to recognize hierarchical watersheds. The inputs are a weighted graph $((V, E), w)$ and a saliency map f. The first step of Algorithm 1 is to compute a lexicographic ordering $\prec$ for (w, f). Then, the binary partition hierarchy $\mathcal{B}$ by $\prec$ and the set of building egdes $E_\prec$ for $\prec$ are computed at lines 2 and 3. Subsequently, the minima of w are obtained at line 4. As established in [74], every minimum of w is a region of $\mathcal{B}$. After computing the set of minima of w, the watershed-cut edges for $\prec$ are obtained at line 5 by browsing the hierarchy $\mathcal{B}$ starting from the leaf regions and by iteratively counting the number of minima included in each region of $\mathcal{B}$. At lines 6-7, we compute the supremum descendant value for $(\prec, f)$ of each building edge for $\prec$. Finally, the last **for** loop (lines $9 - 19$) verifies that the three conditions of Definition 18 for f to be one-side increasing for $\prec$ hold true. The condition 1 of Definition 18 is verified by the two **if** commands between lines 10 and 13. The conditions 2 and 3 of Definition 18 are verified by the **if** commands at lines 14 and 17, respectively. If any of those three conditions is not satisfied, then the algorithm halts and returns **false** and, otherwise, it returns **true**.

Let us now analyse the time complexity of Algorithm 1. Given that the lexicographic ordering for (w, f) can be obtained through the merging sort algorithm, the time complexity of this step is $O(|E|log|E|)$. As established in [74], any binary partition hierarchy can be computed in quasi-linear time with respect to $|E|$ provided that the edges in E are already sorted or can be sorted in linear time. More specifically, the time complexity to compute the binary partition hierarchy $\mathcal{B}$ is $O(|E| \times \alpha(|V|))$, where α is a slowly growing inverse of the single-valued Ackermann function. Having computed the binary partition hierarchy $\mathcal{B}$, the computation of the minima of w and of the watershed-cut edges for $\prec$

can be performed in linear time with respect to $|V|$ as stated in [74]. At lines $6-7$, the supremum descendant values of the building edges for $\prec$ are iteratively computed from the leaves to the root in linear time $O(V)$. Finally, each instruction between lines 10 and 17 can be performed in constant time, which implies that the last **for** loop has a linear time complexity with respect to $|V|$. Therefore, the overall time complexity of Algorithm 1 is $O(|E|log|E|)$.

We illustrate Algorithm 1 in Figure 5.4. The inputs are the weighted graph (G, w) and the saliency map f of Figure 5.4(a) and (b), respectively. We first obtain a lexicographic ordering $\prec$ for (w, f) such that $\{a, b\} \prec \{c, d\} \prec \{e, f\} \prec \{g, h\} \prec \{i, j\} \prec \{a, c\} \prec \{g, i\} \prec \{c, e\} \prec \{d, f\} \prec \{e, g\} \prec \{b, d\} \prec \{f, h\} \prec \{h, j\}$. Then, we obtain the binary partition hierarchy $\mathcal{B}$ by $\prec$, the minima of w (in red) and the four watershed-cut edges of w (underlined) illustrated in Figure 5.4(c). Subsequently, we compute the supremum descendant values for $(\prec, f)$. For each edge u of G, the supremum descendant value of u for $(\prec, f)$ is the greatest value in the set $\{f(v) \mid R_v \subseteq R_u\}$ by Definition 17. We can verify that the range of f is $\{0, 1, 2, 3, 4\}$ and that, among the building edges for $\prec$, all (and only) the watershed-cut edges for $\prec$ have non-zero weights for f. Therefore, the conditions 1 and 2 of Definition 18 for f to be one-side increasing for $\prec$ hold true. Finally, we test the condition 3 of Definition 18. For each watershed-cut edge u of G, we test if $f(u)$ is greater than the supremum descendant value of at least one child of R_u. For the building edges of the regions Y_6, Y_7 and Y_8 the condition 3 holds true, but this is not the case for the region Y_9. Consequently, the map f is not one-side increasing for $\prec$ and Algorithm 1 returns *false*.

3.4 Flattened hierarchical watersheds

In order to compute a hierarchical watershed of (G, w), a sequence of minima of w is often defined by extinction values [98]. When distinct minima of w have the same extinction value, the order between those minima is defined arbitrarily. Let G' be the MSF of (G, w) rooted in the minima of w. By Definition 1, we may say that a hierarchical watershed of (G, w) can be obtained by filtering, one by one, the connected components of G'. Now, let us consider a framework in which the minima with equal extinction values are treated in parallel. In this new framework, the connected components of G' rooted in minima of w with equal extinction values are filtered out simultaneously. We can affirm that the resulting partitions of this framework are a subset of the partitions of a hierarchical watershed of (G, w), and hence a simplified or "flattened" hierarchical watershed.

Algorithm 1 Recognition of hierarchical watersheds

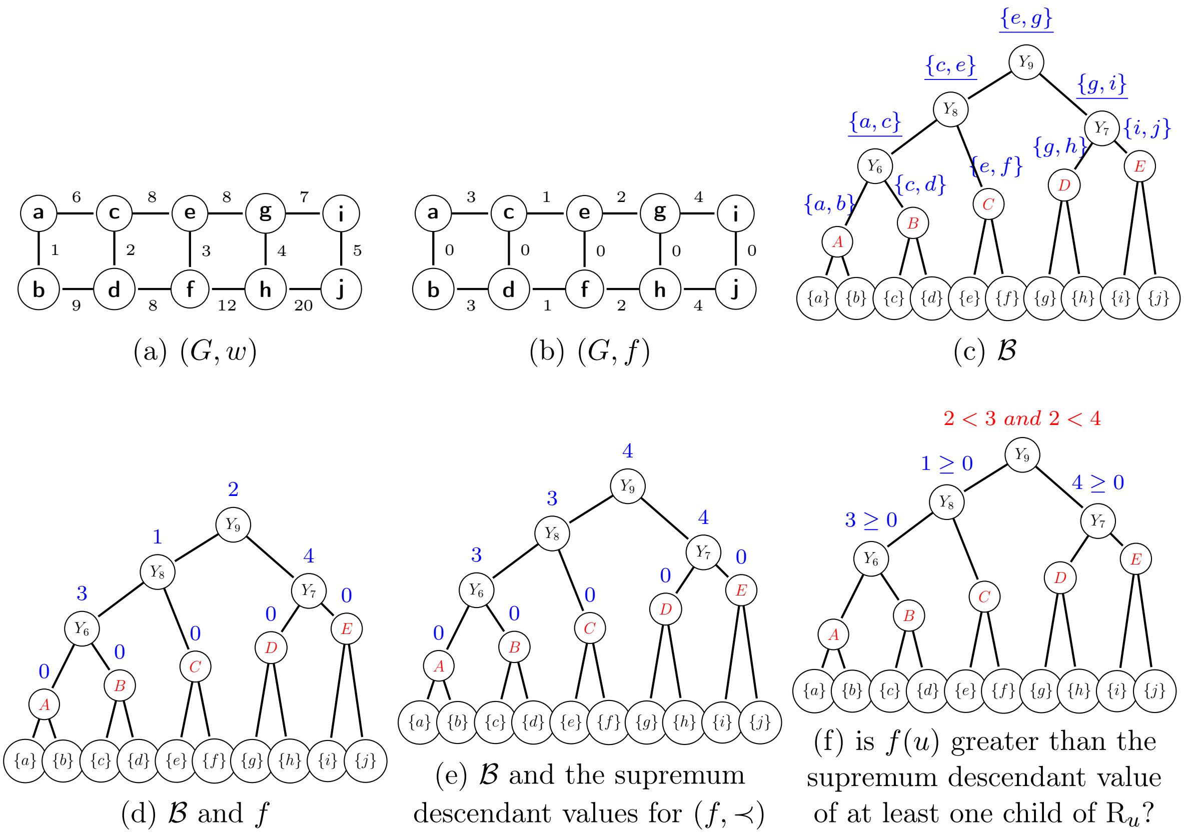
Data: $((V, E), w)$: a weighted graph
f: the saliency map of a hierarchy $\mathcal{H}$ on V
Result: *true* if $\mathcal{H}$ is a hierarchical watershed of (G, w) and *false* otherwise

Compute a lexicographic ordering $\prec$ for (w, f) ▷ $O(|E|log|E|)$
Compute the binary partition hierarchy $\mathcal{B}$ by $\prec$ ▷ $O(|E| \times \alpha(|V|))$ with [74]
Compute the set $E_\prec$ of building edges for $\prec$ ▷ $O(|V|)$
Compute the minima of w ▷ $O(|V|)$ with [74]
Compute the watershed-cut edges for $\prec$ ▷ $O(|V|)$ with [74]
for each building edge u in increasing order for $\prec$ **do** ▷ $O(|V|)$
  $\varphi(u) \leftarrow$ the supremum descendant value of R_u for $(f, \prec)$ ▷ $O(|1|)$
end for
`// Testing of the conditions 1, 2 and 3 of Definition 18 for` f `to be one-side increasing for` $\prec$
for each building edge u in increasing order for $\prec$ **do** ▷ $O(|V|)$
  if $f(u) \notin \{0, 1, \ldots, k\}$ **then** ▷ $O(|1|)$
    return *false* ▷ $O(|1|)$
  end if
  if $f(u) \neq 0$ and $\exists v \in E_\prec$ such that $v \prec u$ and $f(u) = f(v)$ **then** ▷ $O(|1|)$
    return *false* ▷ $O(|1|)$
  end if
  if u is a watershed-cut edge and $f(u) = 0$ or u is not a watershed-cut edge and $f(u) \neq 0$ **then** ▷ $O(|1|)$
    return *false* ▷ $O(|1|)$
  end if
  X and $Y \leftarrow$ children of R_u ▷ $O(|1|)$
  if $\varphi(R_u) < \varphi(X)$ and $\varphi(R_u) < \varphi(Y)$ **then** ▷ $O(|1|)$
    return *false* ▷ $O(|1|)$
  end if
end for
  return *true*

Definition 21 (flattening of hierarchies)**.** *Let $\mathcal{H}$ and $\mathcal{H}'$ be two hierarchies on V such that any partition of $\mathcal{H}$ is a partition of $\mathcal{H}'$. We say that $\mathcal{H}$* is a flattening of *$\mathcal{H}'$.*

Let $\mathcal{H}$ and $\mathcal{H}'$ be two hierarchies on V such that $\mathcal{H}$ is a flattening of $\mathcal{H}'$. If $\mathcal{H}'$ is a hierarchical watershed of (G, w), then we say that $\mathcal{H}$ *is a flattened hierarchical watershed of* (G, w). The reader may note that there can be repeated partitions in both $\mathcal{H}$ and $\mathcal{H}'$. Hence, the hierarchy $\mathcal{H}$ can have more partitions than $\mathcal{H}'$, but $\mathcal{H}$ has less distinct partitions than $\mathcal{H}'$.

We can see that the notion of flattened hierarchical watersheds, even though not formally defined previously, arise naturally in the context of marker-based watershed



(a) (G, w) (b) (G, f) (c) $\mathcal{B}$

(d) $\mathcal{B}$ and f (e) $\mathcal{B}$ and the supremum descendant values for $(f, \prec)$ (f) is $f(u)$ greater than the supremum descendant value of at least one child of R_u?

Figure 3.3: Toy example of Algorithm 1. Given the weighted graphs (G, w) and (G, f), we test if f is the saliency map of a hierarchical watershed of (G, w). We first compute the lexicographic ordering $\prec$ for (w, f) such that $\{a, b\} \prec \{c, d\} \prec \{e, f\} \prec \{g, h\} \prec \{i, j\} \prec \{a, c\} \prec \{g, i\} \prec \{c, e\} \prec \{d, f\} \prec \{e, g\} \prec \{b, d\} \prec \{f, h\} \prec \{h, j\}$. Then, we obtain the binary partition hierarchy $\mathcal{B}_\prec$ by $\prec$, along with the minima of w (in red) and the watershed-cut edges for $\mathcal{B}$ (underlined). Subsequently, we compute the supremum descendant values of the regions of $\mathcal{B}$ for $(f, \prec)$. We may conclude that conditions 1 and 2 of Definition 18 hold true for f, but not the condition 3. Hence, f is not the saliency map of a hierarchical watershed of (G, w).

segmentation. It is noteworthy that, as a hierarchical watershed, all partitions of a flattened hierarchical watershed are optimal in the sense of minima spanning forests.

Among the watershed based hierarchies that are computed without considering a total ordering on the minima, we can cite the waterfall algorithm proposed in [15], which can also be formulated in the framework of weighted graphs. At each step of the waterfall algorithm, several catchment basins of the original image can be merged. In this algorithm, markers are implicitly defined by a sequence of floodings of the original weighted graph.

The following property, whose proof is presented in Appendix 8.1.3, characterizes flattened hierarchical watersheds.

Property 22. *Let $\mathcal{H}$ be a hierarchy on V and let f be the saliency map of $\mathcal{H}$. The hierarchy $\mathcal{H}$ is a flattened hierarchical watershed of (G, w) if and only if there is an altitude ordering $\prec$ for w such that:*

1. *$(V, E_{\prec})$ is a MST of (G, f); and*
2. *for any edge u in $E_{\prec}$, if u is not a watershed-cut edge for $\prec$, then $f(u)$ is zero; and*
3. *for any edge u in $E_{\prec}$, there exists a child R of R_u such that $f(u)$ is greater than or equal to the supremum descendant value of R for $(f, \prec)$.*

We can remark the similarity between Property 22 and Lemma 19, which links hierarchical watersheds to the notion of one-side increasing maps. Let $\mathcal{H}$ be a hierarchy and let f be the saliency map of $\mathcal{H}$. To test if $\mathcal{H}$ is a flattened hierarchical watershed of (G, w), the first condition of Property 22, which is an implication of the first statement of Definition 18, makes sure that we take into account the range of f and not only a subset of the range of f. The second condition of Property 22, which is the forward implication of the second statement of Definition 18, guarantees that the lowest level of $\mathcal{H}$ is equal or coarser than the lowest level of a hierarchical watershed of (G, w). Allied to the second condition of Property 22, the third condition of Property 22, which is equivalent to the third statement of Definition 18, assures that each partition of $\mathcal{H}$ is induced by a MSF rooted in a subset of the set of minima of (G, w).

Algorithm 2 describes our algorithm to recognize flattened hierarchical watersheds, which is very similar to the algorithm to recognize hierarchical watersheds (Algorithm 1). The only difference between algorithms 2 and 1 is that, in Algorithm 2, we do not test if the first condition of Definition 18 holds true, and we test if $(V, E_{\prec})$ is a MST of the input map $((V, E), f)$, where $E_{\prec}$ is the set of building edges for $\prec$. The verification that $(V, E_{\prec})$ is a MST of (G, f) can be done in time $O(|E| log|E|)$ by checking if the sum of the edge weights of a MST of $((V, E), f)$ is equal to the sum of the edge weights of $((V, E_{\prec}), f)$. Hence, the overall time complexity of Algorithm 2 is the same of Algorithm 1: $O(|E| log|E|)$.

3.5 Conclusion

In this chapter, we proposed a solution the problem of recognition of hierarchical watersheds. We introduced a characterization of hierarchical watersheds and, based on this

Algorithm 2 Recognition of flattened hierarchical watersheds

Data: $((V, E), w)$: a weighted graph
f: the saliency map of a hierarchy $\mathcal{H}$ on V
Result: *true* if $\mathcal{H}$ is a flattened hierarchical watershed of (G, w) and *false* otherwise
/* **Lines** $1 - 8$ **of Algorithm 1** */ ▷ $O(|E|log|E|)$
`// Testing of the conditions 1, 2 and 3 of Property 22`
`for` f `to be a flattened hierarchical watershed of` $((V, E), w)$
if $(V, E_{\prec})$ is not a MST of $((V, E), f)$ **then** ▷ $O(|E|log|E|)$
  return *false*
end if
for each building edge u in increasing order for $\prec$ **do** ▷ $O(|V|)$
  if u is not a watershed-cut edge and $f(u) \neq 0$ **then** ▷ $O(|1|)$
    return *false* ▷ $O(|1|)$
  end if
  X and $Y \leftarrow$ children of R_u ▷ $O(|1|)$
  if $\varphi(R_u) < \varphi(X)$ and $\varphi(R_u) < \varphi(Y)$ **then** ▷ $O(|1|)$
    return *false* ▷ $O(|1|)$
  end if
end for
return *true*

characterization, we designed an efficient algorithm to determine if a hierarchy is a hierarchical watershed of any given weighted graph. To consider the hierarchies that are obtained by a partial ordering on the minima of a weighted graph, we introduced the notion of flattened hierarchical watersheds, which is a relaxed definition of hierarchical watersheds.

Later, in Chapter 7, we study a method to combine hierarchies through their saliency maps and show experimental results on combinations of hierarchical watersheds [57]. Those results raise the question of whether the resulting combinations are hierarchical watersheds or flattened hierarchical watersheds. In the affirmative case, we could infer that there exists an increasing attribute A (or several such attributes) such that the combinations of hierarchical watersheds are precisely the hierarchical watersheds based on A. This problem is addressed in details in Chapter 7, where we present experimental results with our algorithm to recognize (flattened) hierarchical watersheds applied to the combinations of hierarchical watersheds assessed in [57].

Chapter 4

Watersheding hierarchies

In this chapter, we introduce an operator, called watersheding, that converts any hierarchy of partitions into a hierarchical watershed of a given weighted graph. The results presented here were introduced in the following article:

- D. S. Maia, J. Cousty, L. Najman, and B. Perret. Watersheding hierarchies. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 124–136. Springer, 2019.

4.1 Introduction

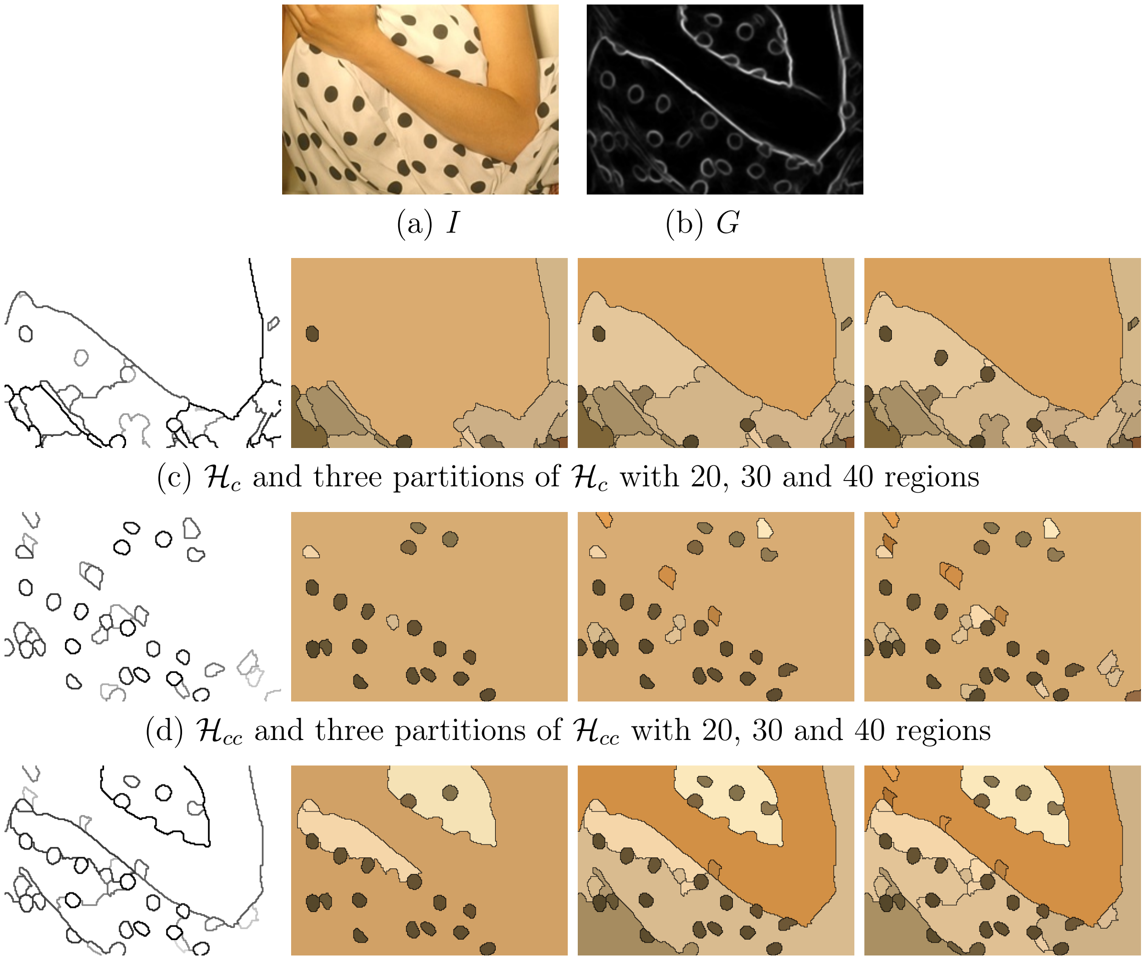
In the context of image segmentation, hierarchies (of partitions) are sequences of nested partitions of image pixels. At the highest levels of a hierarchy, we have the most representative regions according to a given criterion, such as size and contrast. As discussed in Section 2.4, hierarchies can be equivalently represented by saliency maps, in which the contours between regions are weighted according to their level of disappearance in the hierarchy. Thank to the bijection between hierarchies and saliency maps [25], we work indifferently with any of those notions in this study.

In mathematical morphology, hierarchies are often obtained from the watershed transform [22, 14]. With the definitions proposed in [22, 74, 27] (see Definition 1), hierarchical watersheds are optimal in the sense of minimum spanning forests. Furthermore, efficient algorithms to compute those hierarchies have been designed [24, 74]. Moreover, watersheds can be linked to a broader family of combinatorial optimization problems, such as random walkers and graph cuts, as demonstrated in [20].

In this study, we propose the *watersheding operator*, which, given an edge-weighted graph (G, w), transforms (the saliency map of) any hierarchy connected for G into (the saliency map of) a hierarchical watershed of (G, w). Figure 4.1 illustrates an application of the watersheding operator. The goal is to obtain, from the image gradient G, a hierarchical watershed of G that highlights the most perceptually significant regions of the image I of Figure 4.1(a), including the circular ones. A straightforward method to achieve this goal is to compute the hierarchical watershed of G based on regularized circularity attribute values (see Section 2.7.3). The hierarchy $\mathcal{H}_c$ of Figure 4.1(c) is a hierarchical watershed of G based on circularity values after regularization with the max-rule. We can observe that the hierarchy $\mathcal{H}_c$ does not succeed at highlighting all the main circular regions of the image I. Alternatively, we computed another hierarchy $\mathcal{H}_{cc}$, illustrated in Figure 4.1(d), by simply weighing each contour of the regions of the area-based hierarchical watershed of G with the maximum circularity value among the regions that share this contour. The hierarchy $\mathcal{H}_{cc}$ brings to the fore the main circular regions of I, however it is not a hierarchical watershed of G. By applying the watersheding operator on $\mathcal{H}_{cc}$, we obtain the hierarchiy $\mathcal{H}_w$ of Figure 4.1(d). Analogous to the hierarchy $\mathcal{H}_c$, the result of the watersheding operator $\mathcal{H}_w$ is also a hierarchical watershed of G. We can see that the circular regions of the image I are better highlighted in the result of the watersheding operator $\mathcal{H}_w$ when compared to the straightforward approach $\mathcal{H}_c$. Furthermore, besides highlighting the main circular regions present at high levels of the hierarchy $\mathcal{H}_{cc}$, the hierarchy $\mathcal{H}_w$ brings to the fore the region covering the arm, which is also a perceptually significant region.

We present the formal definition of the watersheding operator along with its main properties in Section 4.2. We show that the watersheding operator is idempotent and that the saliency maps of hierarchical watersheds are the fixed points of this operator. Then, we conclude in Theorem 33 that the watersheding operator also provides a characterization of hierarchical watersheds and, hence, is an alternative solution to the problem of recognizing hierarchical watersheds discussed in Chapter 3. In Section 4.3, we propose an efficient algorithm that implements the watersheding operator. Finally, we discuss applications of this operator in Section 4.4.

Important notations: in the remainder of this chapter, the symbol (G, w) denotes a weighted graph whose vertex set is connected. For the sake of simplicity, we consider that G is a tree. To shorten the notation, the vertex set of G is denoted by V and its edge set is denoted by E. Without loss of generality, we also assume that the range of w is included in the set $\mathbb{E}$ of all integers from 0 to $|E| - 1$ (otherwise, one could always consider an increasing one-to-one correspondence from the set $\{w(u) \mid u \in E\}$ into the subset $\{0, ..., |\{w(u) \mid u \in E\}| - 1\}$ of $\mathbb{E}$). Every hierarchy considered in this chapter is connected for G and therefore, for the sake of simplicity, we use the term *hierarchy* instead



(a) I (b) G

(c) $\mathcal{H}_c$ and three partitions of $\mathcal{H}_c$ with 20, 30 and 40 regions

(d) $\mathcal{H}_{cc}$ and three partitions of $\mathcal{H}_{cc}$ with 20, 30 and 40 regions

(e) $\mathcal{H}_w$ and three partitions of $\mathcal{H}_w$ with 20, 30 and 40 regions

Figure 4.1: (a) an image I. (b) the gradient G of I computed with the edge detector introduced in [29]. (c) From left to right: the saliency map of the hierarchical watershed $\mathcal{H}_c$ of G based on a regularized circularity attribute, and three partitions of $\mathcal{H}_c$ containing 20, 30 and 40 regions, respectively. (d) From left to right: the saliency map of a circularity based hierarchy $\mathcal{H}_{cc}$, and three partitions of $\mathcal{H}_{cc}$ containing 20, 30 and 40 regions, respectively. (e) the watersheding $\mathcal{H}_w$ of the saliency map of $\mathcal{H}_{cc}$ (for G), and three partitions of $\mathcal{H}_w$ containing 20, 30 and 40 regions, respectively.

of *hierarchy which is connected for* G. We denote by n the number of minima of w. Every sequence of minima of w considered in this chapter is a sequence of n pairwise distinct minima of w and, then we use the term *sequence of minima of* w instead of *sequence of* n *pairwise distinct minima of* w. By abuse of terminology, when no confusion is possible, if M is a minimum of w, we call the set $V(M)$ of vertices of M as a minimum of w.

4.2 Watersheding operator

In this section, we introduce the watersheding (operator), which maps any saliency map into the saliency map of a hierarchical watershed of (G, w). To ease the reading of this section, the proof of part of the properties stated here are delayed to Appendix 8.2.

The idea underlying the watersheding operator is to invert the method to compute hierarchical watersheds proposed in [74, 27] and revised in Section 2.6.1. As the methods proposed in [74, 27], the watersheding operator relies on the notion of binary partition hierarchy (by altitude ordering) discussed in Section 2.5.2. In order to present the watersheding operator, we first remind the definition of extinction values for a sequence of minima.

Important notation: given an altitude ordering $\prec$ for w and a building edge u for $\prec$ (see definitions in Section 2.5.2), we denote by R_u the region of the binary partition hierarchy $\mathcal{B}_\prec$ by $\prec$ whose building edge is u. The set of building edges for $\prec$ is denoted by $E_\prec$.

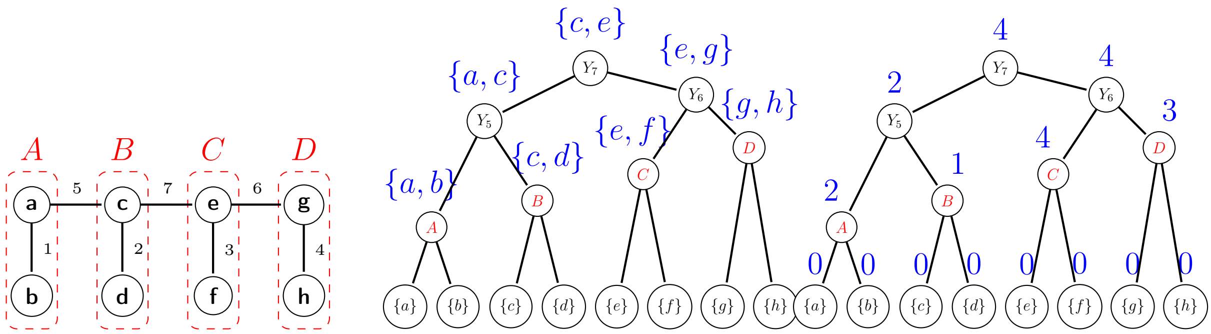
Let $\prec$ be an altitude ordering for w, let $\mathcal{S} = (M_1, \ldots, M_n)$ be a sequence of minima of w and let R be any region of the binary partition hierarchy $\mathcal{B}_\prec$ by $\prec$. By Definition 3, the extinction value of R for $(\mathcal{S}, \prec)$ is zero if there is no minimum of w included in R and, otherwise, it is the maximum value i in $\{1, \ldots, n\}$ such that the minimum M_i is included in R. Let ϵ be a map from the regions of $\mathcal{B}_\prec$ into $\mathbb{R}$ such that, for any region R of $\mathcal{B}_\prec$, the value $\epsilon(R)$ is the extinction value of R for $(\mathcal{S}, \prec)$. We say that *ϵ is the extinction map for $(\mathcal{S}, \prec)$*, that *ϵ is an extinction map for $\mathcal{S}$* and that *ϵ is an extinction map for $\prec$*.

The following property, whose proof is given in Appendix 8.2.1, characterizes extinction maps.

Property 23. *Let $\prec$ be an altitude ordering for w and let ϵ be a map from the regions of $\mathcal{B}_\prec$ into $\mathbb{R}$. The map ϵ is an* extinction map for $\prec$ *if and only if the following statements hold true:*

1. *$\{\epsilon(R) \mid R$ is a region of $\mathcal{B}_\prec\} = \{0, \ldots, n\}$;*
2. *for any two distinct minima M_1 and M_2 of w, we have $\epsilon(M_1) \neq \epsilon(M_2)$; and*
3. *for any region R of $\mathcal{B}_\prec$, the value $\epsilon(R)$ is equal to $\vee\{\epsilon(M)$ such that M is a minimum of w included in $R\}$.*

We provide an example of an extinction map in Figure 4.2. Let (G, w) be the graph of Figure 4.2(a), let $\mathcal{B}_\prec$ be the binary partition hierarchy of Figure 4.2(b) for the unique



(a) (G, w) (b) $\mathcal{B}_{\prec}$ (c) ϵ

Figure 4.2: (a) a weighted graph (G, w). (b) the binary partition hierarchy $\mathcal{B}_{\prec}$ for the unique altitude ordering $\prec$ for w. (c) the extinction map ϵ for the sequence (B, A, D, C) and $\prec$.

altitude ordering $\prec$ for w, and let ϵ be the map from the regions of $\mathcal{B}_{\prec}$ into $\mathbb{R}$ illustrated in Figure 4.2(c). We can see that the map ϵ is the extinction map for $\prec$ and for the sequence (B, A, D, C) of minima of w.

The next property, whose proof is presented in Appendix 8.2.2, clarifies the relation between hierarchical watersheds and extinction maps. As established in [27], given a sequence $\mathcal{S}$ of minima of w, we can compute the saliency map of a hierarchical watershed for $\mathcal{S}$ by considering any extinction map for $\mathcal{S}$. Since the edge weights of w are not necessarily pairwise distinct, given any sequence $\mathcal{S}$ of minima of w, there might be several distinct hierarchical watersheds of (G, w) for $\mathcal{S}$. Let $\mathcal{S}$ be a sequence of minima of w. As established in the following property, we can associate any hierarchical watershed $\mathcal{H}$ of (G, w) for $\mathcal{S}$ with an altitude ordering $\prec$ for w such that, for any building edge u for $\prec$, the weight of u for the saliency map $\Phi(\mathcal{H})$ is obtained from the extinction map for $(\mathcal{S}, \prec)$.

Property 24. *Let $\mathcal{H}$ be a hierarchy. The hierarchy $\mathcal{H}$ is a hierarchical watershed of (G, w) if and only if there exists an altitude ordering $\prec$ for w and an extinction map ϵ for $\prec$ such that:*

1. *$(V, E_{\prec})$ is a MST of $(G, \Phi(\mathcal{H}))$; and*
2. *for any edge u in $E_{\prec}$, the value $\Phi(\mathcal{H})(u)$ is equal to $\min\{\epsilon(R)$ such that R is a child of $R_u\}$.*

Let $\mathcal{S}$ be a sequence of minima of w and let $\prec$ be an altitude ordering for w. As established in Section 2.6.1, the saliency map of a hierarchical watershed of (G, w) for $\mathcal{S}$ can be obtained through the following steps:

1. computation of the binary partition hierarchy $\mathcal{B}_{\prec}$ by $\prec$;

2. computation of the extinction map ϵ for $(\mathcal{S}, \prec)$;

3. computation of the persistence values for $(\mathcal{S}, \prec)$ (see Definition 4); and

4. computation of the saliency map f of the hierarchy induced by $(\mathcal{S}, \prec)$ which, by Property 6, is the saliency map of a hierarchical watershed of (G, w) for $\mathcal{S}$. Since G is a tree, any edge of G is a building edge for $\prec$. Hence, for any edge u in E, the value $f(u)$ is the persistence value of u for $(\mathcal{S}, \prec)$.

Inspired by this simple and efficient method to compute hierarchical watersheds, we propose the watersheding operator. Given an altitude ordering $\prec$ for w and a map f, we first find an *approximated extinction map* ϵ such that, if f is the saliency map of a hierarchical watershed of (G, w), then the map ϵ is an extinction map for $\prec$ and f is the saliency map induced by ϵ. Then, we define the *estimated sequence* $\mathcal{S}$ *of minima* of w ordered in increasing order for ϵ. The watersheding of f is then defined by the persistence values for $(\mathcal{S}, \prec)$.

To introduce approximated extinction maps, we first review the notion of supremum descendant map introduced in Chapter 3, and then we introduce the auxiliary notions of non-leaf ordering and dominant region.

Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$. Let R be a region of the binary partition hierarchy $\mathcal{B}_{\prec}$ by $\prec$. By Definition 17, the supremum descendant value of R for $(f, \prec)$ is $\vee\{f(v) \mid v \in E, R_v \subseteq R\}$.

Important notation: to lighten the notation, in the remainder of this chapter, the supremum descendant value of a region is simply called the descendant value of this region.

Definition 25 (non-leaf ordering)**.** *Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$. The* non-leaf ordering for $(f, \prec)$ *is the total ordering $\ll$ on the building edges for $\prec$, such that, for any two building edges u and v for $\prec$, we have $u \ll v$ if either the descendant value of R_u (for $(f, \prec)$) is strictly lower than the descendant value of R_v, or the descendant values of R_u and R_v are equal and $u \prec v$.*

Definition 26 (dominant region)**.** *Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$. Let $\ll$ be the non-leaf ordering for $(f, \prec)$. Let R be a non-leaf region of $\mathcal{B}_{\prec}$ different from V. Let u and v be the building edges of R and of the sibling of R, respectively. We say that R is* a dominant region for $(f, \prec)$ *if:*

1. there is a minimum of w included in R; and

2. *either:*

   - $v \ll u$; *or*
   - *there is no minimum of w included in the sibling of R.*

For instance, let (G, w) be the weighted graph shown in Figure 4.2(a) and let $\prec$ be the unique altitude ordering for w. Let $\mathcal{B}_\prec$ be the binary partition hierarchy by $\prec$ shown in Figure 4.2(b), and let f be the map of Figure 4.3(a). The descendant values for $(f, \prec)$ are depicted in Figure 4.3(b). Let $\ll$ be the non-leaf ordering for $(f, \prec)$ such that $\{a, b\} \ll \{c, d\} \ll \{e, f\} \ll \{g, h\} \ll \{a, c\} \ll \{c, e\} \ll \{e, g\}$. The dominant regions of $\mathcal{B}_\prec$ for $(f, \prec)$ are the regions B, D and Y_6 (in bold).

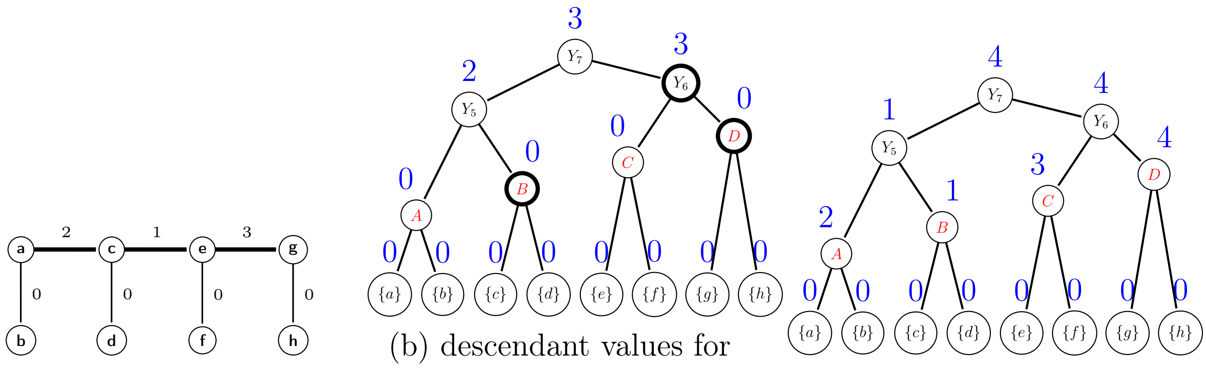
Let f and g be two maps from E into $\mathbb{R}$. A *lexicographic ordering for* (f, g) is a total ordering $\prec$ on E such that, for any two edges u and v in E, we have $u \prec v$ if $f(u) < f(v)$ or if $f(u) = f(v)$ and $g(u) \leq g(v)$. We can note that any lexicographic ordering for (f, g) is an altitude ordering for f.

Definition 27 (approximated extinction map)**.** *Let f be a map from E into $\mathbb{R}$ and let $\prec$ be a lexicographic ordering for (w, f). The* approximated extinction map for $(f, \prec)$ *is the map ξ from the set of regions of $\mathcal{B}_\prec$ into $\mathbb{R}$ such that:*

1. *$\xi(R) = k + 1$ if R is the vertex set V of G, where k is the descendant value of R for $(f, \prec)$; and*
2. *$\xi(R) = \xi(parent(R))$ if R is a dominant region for $(f, \prec)$; and*
3. *$\xi(R) = f(u)$, where u is the building edge of the parent of R, otherwise.*

Let f be the map of Figure 4.3(a) and let $\prec$ be the unique altitude ordering for the map w of Figure 4.2(a). From Definition 27, we deduce that the approximated extinction map for $(f, \prec)$ can be obtained in a recursive top-down fashion: approximated extinction values are propagated from the root of the binary partition hierarchy $\mathcal{B}_\prec$ to the leaf regions of $\mathcal{B}_\prec$. Along this recursive process, the approximated extinction value of every region R of $\mathcal{B}_\prec$ is propagated to at most one of the children of R: a child of R that is a dominant region for $(f, \prec)$. In Figure 4.3(c), we show the approximated extinction map ξ for $(f, \prec)$. The next property, whose proof is detailed in Appendix 8.2.3, establishes that ξ is an extinction map if and only if f is the saliency map of a hierarchical watershed of (G, w).

Theorem 28. *Let f be a map from E into $\mathbb{R}$, let $\prec$ be a lexicographic ordering for (w, f), and let ξ be the approximated extinction map for $(f, \prec)$. The map f is the saliency map of a hierarchical watershed of (G, w) if and only if the map ξ is an extinction map for $\prec$.*



(a) (G, f) (b) descendant values for $(f, \prec)$ (c) ξ

Figure 4.3: (a) a weighted graph (G, f). (b) the descendant value for $(f, \prec)$, where $\prec$ is the altitude ordering for the map w of Figure 4.2(a). The dominant regions for $(f, \prec)$ are depicted in bold. (c): the approximated extinction map ξ for $(f, \prec)$.

By Property 23, we can conclude that the map ξ of Figure 4.3(c) is not an extinction map for $\prec$ because $\vee\{\xi(A), \xi(B)\} = 2$ is different from $\xi(Y_5) = 1$, which contradicts the Property 23 (statement 3) on extinction maps. By Theorem 28, we may conclude that the map f of Figure 4.3(a) is not the saliency map of a hierarchical watershed of the graph (G, w) of Figure 4.2(a).

In the next definition, we introduce estimated sequences of minima obtained through approximated extinction maps.

Definition 29 (estimated sequence of minima)**.** *Let f be a map from E into $\mathbb{R}$, let $\prec$ be a lexicographic ordering for (w, f), and let ξ be the approximated extinction map for $(f, \prec)$. Let $\ll$ be the non-leaf ordering for $(f, \prec)$. The estimated sequence of minima (of w) for $(f, \prec)$ is the sequence $(M_1, \ldots, M_n)$ such that, for any i and j in $\{1, \ldots, n\}$, if $i < j$, then either:*

- *$\xi(M_i) < \xi(M_j)$; or*
- *$\xi_f(M_i) = \xi_f(M_j)$ and $M_i \ll M_j$.*

For instance, in Figure 4.3(c), we can see that $\xi(B) < \xi(A) < \xi(C) < \xi(D)$. Therefore, the estimated sequence of minima for $(f, \prec)$ is $\mathcal{S} = (B, A, C, D)$. The next property establishes that, if f is the saliency map of a hierarchical watershed of (G, w), then f is the saliency map of a hierarchical watershed of (G, w) for $\mathcal{S}$.

Property 30. *Let f be a map from E into $\mathbb{R}$, let $\prec$ be a lexicographic ordering for (w, f), and let $\mathcal{S}$ be the estimated sequence of minima for $(f, \prec)$. If f is the saliency map of a*

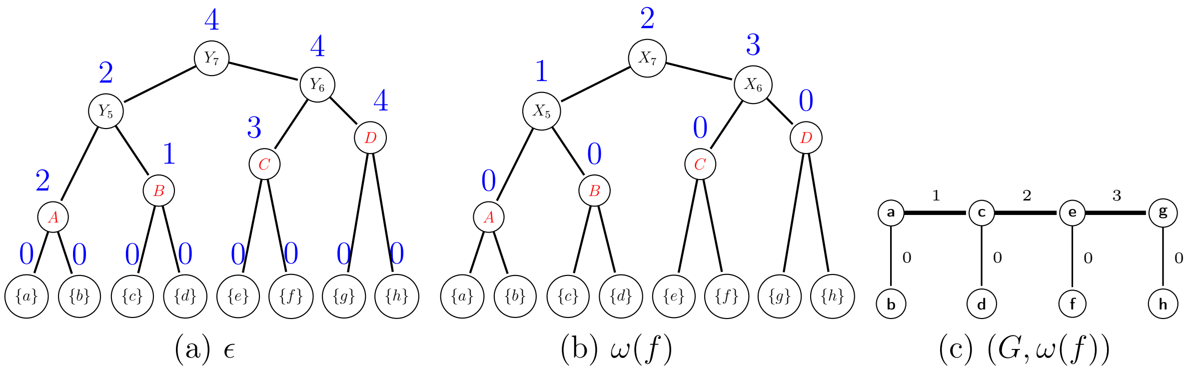


Figure 4.4: (a) the extinction map ϵ for the estimated sequence of minima $\mathcal{S} = (B, A, C, D)$ for f and $\prec$, where f is the map of Figure 4.3(a) and $\prec$ is the unique altitude ordering for the map w of Figure 4.2(a). (b) the watersheding $\omega(f)$ of f represented on the hierarchy $\mathcal{B}_\prec$ of 4.3(b). The weight above each region R of $\mathcal{B}_\prec$ is the weight of the building edge of R for $\omega(f)$. (c) the watersheding of f depicted on the graph G.

hierarchical watershed of (G, w)*, then* f *is the saliency map of a hierarchical watershed of* (G, w) *for* $\mathcal{S}$*.*

The proof of Property 30 is given in Appendix 8.2.4.

Having defined approximated extinction maps and estimated sequences of minima, we formalize the watersheding operator in the following definition.

Definition 31 (watersheding)**.** *Let* f *be a map from* E *into* $\mathbb{R}$*, let* $\prec$ *be a lexicographic ordering for* (w, f)*, and let* $\mathcal{S}$ *be the estimated sequence of minima for* $(f, \prec)$*. Let* ϵ *be the extinction map for* $(\mathcal{S}, \prec)$*. The* watersheding of f (for $\prec$) *is the map* $\omega(f)$ *from* E *into* $\mathbb{R}$ *such that, for any edge* u*, the value* $\omega(f)(u)$ *is the persistence value of* u *for* $(\mathcal{S}, \prec)$*,* i.e.,*:*

$$\omega(f)(u) = min\{\epsilon(R) \mid R \text{ } is \text{ } a \text{ } child \text{ } of \text{ } R_u\}.$$

In Figure 4.4(b) and (c), we show the watersheding $\omega(f)$ of the map f of Figure 4.3(a). After obtaining the estimated sequence of minima $\mathcal{S} = (B, A, C, D)$ for $(f, \prec)$, we compute the extinction map ϵ for $(\mathcal{S}, \prec)$. Then, the watersheding of f is obtained according to Definition 31. We can verify that $\omega(f)$ is the saliency map of a hierarchical watershed of (G, w) for $\mathcal{S}$.

In the following theorem, we establish that the watersheding of any map is the saliency map of a hierarchical watershed of (G, w).

Theorem 32. *Let* f *be a map from* E *into* $\mathbb{R}$ *and let* $\prec$ *be a lexicographic ordering for* (w, f)*. Let* $\mathcal{S}$ *be the estimated sequence of minima for* f *and* $\prec$*. The watersheding* $\omega(f)$ *of* f *(for* $\prec$*) is a saliency map of a hierarchical watershed of* (G, w) *for* $\mathcal{S}$*.*

Proof. Let ϵ be the extinction map for $(\mathcal{S}, \prec)$. By Definition 31, for any edge u, the value $f(u)$ is the persistence value of u for $(\mathcal{S}, \prec)$. By Definition 5, we conclude that $\mathcal{QFZ}(G, f)$ is the hierarchy induced by $(\mathcal{S}, \prec)$. Then, by Property 6, the hierarchy $\mathcal{QFZ}(G, f)$ is a hierarchical watershed for $\mathcal{S}$. Hence, since G is a tree, the map f is the saliency map of the hierarchical watershed of (G, w) for $\mathcal{S}$. □

The following theorem, whose proof is given in Appendix 8.2.5, establishes that the watersheding operator is idempotent, that the saliency maps of the hierarchical watersheds of (G, w) are the fixed points of the watersheding operator, and that the watersheding operator provides a characterization of hierarchical watersheds. The later statement implies that the watersheding operator offers an alternative solution to the problem of recognizing hierarchical watersheds studied in Chapter 3.

Theorem 33. *Let $\mathcal{H}$ be a hierarchy, let f be the saliency map of $\mathcal{H}$ and let $\prec$ be a lexicographic ordering for (w, f). The following statements hold true:*

1. *The hierarchy $\mathcal{H}$ is a hierarchical watershed of (G, w) if and only if the watersheding $\omega(f)$ of f (for $\prec$) is equal to f.*
2. *The watersheding $\omega(f)$ of f is the saliency map of a hierarchical watershed of (G, w).*
3. *The watersheding $\omega(\omega(f))$ of $\omega(f)$ is equal to $\omega(f)$.*

4.3 Watersheding operator algorithm

In this section, we present an efficient algorithm to compute the watersheding of any map following Definition 31. Algorithm 3 provides a description of our watersheding (operator) algorithm. The inputs are a weighted tree $((V, E), w)$ and a map f from E into $\mathbb{R}$. The first step of Algorithm 1 is to compute a lexicographic ordering $\prec$ for (w, f). Then, the binary partition hierarchy $\mathcal{B}$ by $\prec$ is computed at line 2 with the method proposed in [74]. Subsequently, the minima of w are obtained at line 3. As established in [74], every minimum of w is a region of $\mathcal{B}$. After computing the set of minima of w, the descendant values for $(f, \prec)$ are obtained at lines 4-5. For each building edge u for $\prec$, by Definition 17, the descendant value of u for $(f, \prec)$ is the maximal value $f(v)$ such that the region R_v is a subset of the region R_u. Then, the non-leaf ordering $\ll$ for $(f, \prec)$ is obtained at line 7. At line 8, the dominant regions for $(f, \prec)$ are computed in a single pass on the hierarchy $\mathcal{B}$. Then, the **for** loop at lines 9-20 computes the approximated extinction map ξ for $(f, \prec)$ in a top-down fashion following Definition 27. Subsequently, at line 20, the estimated sequence of minima $\mathcal{S}$ for $(f, \prec)$ is computed by ordering the minima of w in increasing order of their values in ξ. At lines 22-23, the extinction and persistence values for $(\mathcal{S}, \prec)$ are computed from the leaf regions to the root of $\mathcal{B}$. Finally, the watersheding ω of f for $\prec$ is computed at lines 24-25. The weight of each edge u in E is assigned to the persistence value of u. Then, Algorithm 3 returns the map ω.

Let us now analyze the time complexity of Algorithm 3. Given that the lexicographic ordering for (w, f) can be obtained through the merging sort algorithm, the time complexity of this step is $O(|E|log|E|)$. As established in [74], any binary partition hierarchy can be computed in quasi-linear time with respect to $|E|$ provided that the edges in E are already sorted or can be sorted in linear time. More specifically, the time complexity to compute the binary partition hierarchy $\mathcal{B}$ is $O(|E| \times \alpha(|V|))$, where α is a slowly growing inverse of the single-valued Ackermann function. Having computed the binary partition hierarchy $\mathcal{B}$, the computation of the minima of w can be performed in linear time with respect to $|V|$ as stated in [74]. At lines 4-5, the descendant values are iteratively computed from the leaf regions to the root of $\mathcal{B}$ in linear time $O(|V|)$. Then, the non-leaf ordering for $(f, \prec)$ can be obtained using the merging sort algorithm in time $O(|V|log|V|)$. The approximated extinction map for $(f, \prec)$ is computed in one pass over the regions of $\mathcal{B}$ and, hence, in linear time $O(|V|)$. Given that the minima of w are ordered using the merging sort algorithm at line 30, the time complexity to obtain the estimated sequence of minima for $(f, \prec)$ is $O(|V|log|V|)$. Then, the extinction and persistence values for $(\mathcal{S}, \prec)$ can be computed recursively from the leaf regions to the root of $\mathcal{B}$ in linear time $O(|V|)$. Therefore, the overall time complexity of Algorithm 3 is $O(|E|log|E|)$.

4.4 Illustrations of applications in image analysis

In this chapter, we introduced the watersheding operator, which converts any map into the saliency map of a hierarchical watershed of (G, w). In the following, we illustrate two possible applications of the watersheding operator in image analysis.

- *Regularization of hierarchies based on non-increasing attributes.* In a hierarchical watershed, the order in which regions (catchment basins) are merged are often defined by extinction values associated with increasing regional attributes, such as area and volume [70]. To compute a hierarchical watershed of (G, w) based on a attribute A, we first obtain extinction values based on A [98]. Then, we compute

Algorithm 3 Watersheding operator

Data: $((V, E), w)$: a weighted graph
f: the saliency map of a hierarchy $\mathcal{H}$ on V
Result: the watersheding of f

Compute a lexicographic ordering $\prec$ for (w, f) ▷ $O(|E|log|E|)$
Compute the binary partition hierarchy $\mathcal{B}$ by $\prec$ ▷ $O(|E| \times \alpha(|V|))$ with [74]
Compute the minima of w ▷ $O(|V|)$ with [74]
for each building edge u in increasing order for $\prec$ **do** ▷ $O(|V|)$
    $\varphi(u) \leftarrow$ the descendant value of R_u for $(f, \prec)$ ▷ $O(|1|)$
end for
Compute the non-leaf ordering $\ll$ for $(f, \prec)$ (Def. 25) ▷ $O(|V|log|V|)$
Compute the dominant regions of $\mathcal{B}$ for $(f, \prec)$ (Def. 26) ▷ $O(|V|)$
`// Computation of the approximated extinction map for` $(f, \prec)$
for each building edge u in decreasing order for $\prec$ **do** ▷ $O(|V|)$
    if u is the building edge of V **then** ▷ $O(|1|)$
        $\xi(u) \leftarrow$ the descendant value of V ▷ $O(|1|)$
    else
        $v \leftarrow$ building edge of the parent of R_u ▷ $O(|1|)$
        if R_u is a dominant region for $(f, \prec)$ **then** ▷ $O(|1|)$
            $\xi(R_u) \leftarrow \xi(R_v)$ ▷ $O(|1|)$
        else
            $\xi(R_u) \leftarrow f(v)$ ▷ $O(|1|)$
        end if
    end if
end for
$\mathcal{S} \leftarrow$ sequence of minima ordered in increasing order for ξ ▷ $O(|V|log|V|)$
Compute the extinction values for $(\mathcal{S}, \prec)$ (Def. 3) ▷ $O(|V|)$
Compute the persistence values for $(\mathcal{S}, \prec)$ (Def. 4) ▷ $O(|V|)$
for each building edge u for $\prec$ **do** ▷ $O(|V|)$
    $\omega(u) \leftarrow$ persistence value of u for $(\mathcal{S}, \prec)$ ▷ $O(|1|)$
end for
return ω

the hierarchical watershed $\mathcal{H}$ of (G, w) for the sequence $\mathcal{S}$ of minima w ordered by their extinction values. The hierarchy $\mathcal{H}$ corresponds to a sequence of filterings of the watershed of (G, w) in which the least important regions according to A are the first regions to be suppressed. When dealing with non-increasing attributes, *e.g.* circularity and perimeter, we can obtain extinction values by applying a regularization rule [88] on the attribute values (see Section 2.7.3). However, computing a hierarchical watershed using a non-increasing criterion does not guarantee that the most relevant regions according to this criterion are preserved at the highest levels of the hierarchy. Alternatively, instead of computing hierarchical watersheds from regularized attribute values, we can compute the watersheding of any hierarchy based on a non-increasing attribute. This is illustrated in Figures 4.5 and 4.6. For each image gradient, we show the saliency maps of three hierarchies: a hierarchical watershed based on regularized circularity attribute values, a circularity based hierarchy that is not a hierarchical watershed (see Section 2.7.3), and the watersheding of the latter hierarchy. We can see that, in all cases, in the saliency maps resulting from the watersheding operator, the circular regions are more highlighted when compared to the hierarchical watershed computed from regularized circularity values.

- *Refinement of coarse hierarchies.* In [60], the authors propose a high-quality method (COB) to compute hierarchies. However, in some cases, fine regions are not included in the resulting hierarchies. This is the case of the hierarchies of Figures 4.7 and 4.8. In the saliency map of the COB hierarchy of Figure 4.7, the region of the lips do not appear even at the lowest levels of the hierarchy and, in the saliency map of the COB hierarchy of Figure 4.8, spurious regions in the background appear at higher levels than the eyes region. The watersheding of those hierarchies improve the segmentation of the face features of both images while taking into consideration coarse levels of the initial COB hierarchies.

4.5 Conclusion

We introduced an idempotent operator, called watersheding, which converts (the saliency map of) any hierarchy into (the saliency map of) an hierarchical watershed of (G, w). We presented an efficient algorithm to compute the watersheding of a map, and two potential applications of the watersheding, namely the computation of hierarchical watersheds based on non-increasing attributes and the refinement of coarse hierarchies.

As future work, we aim to investigate a relevant question regarding the watersheding operator: given a map f, how "close" is the watersheding of f to this map f? More formally, does the watersheding solve the problem of finding a hierarchical watershed that better approximates a hierarchy in the sense of a well defined objective function (*e.g.* the Gromov-Hausdorff distance between hirarchies analysed in [34])?

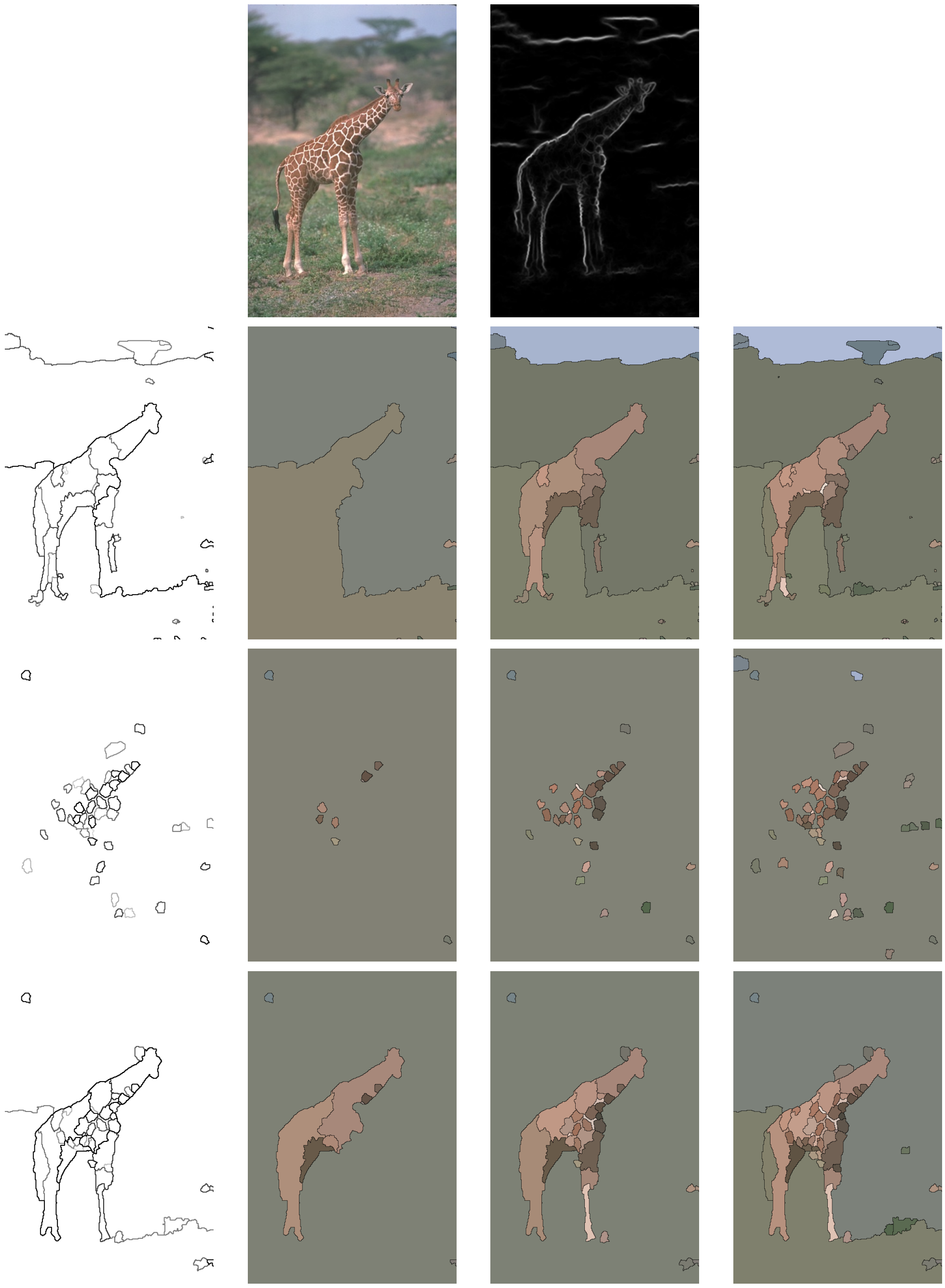


Figure 4.5: First line from left to right: original image I and the gradient G of I computed using the edge detector introduced in [29]. Second line from left to right: the saliency map of the hierarchical watershed $\mathcal{H}_c$ of G based on regularized circularity attribute values and three partitions of $\mathcal{H}_c$ with 10, 35 and 60 regions, respectively. Third line from left to right: the saliency map of the circularity based hierarchy $\mathcal{H}_{cc}$, which is not a hierarchical watershed of G, and three partitions of $\mathcal{H}_{cc}$ with 10, 35 and 60 regions, respectively. Fourth line: the watersheding $\mathcal{H}_w$ of $\mathcal{H}_{cc}$ and three partitions of $\mathcal{H}_w$ with 10, 35 and 60 regions, respectively.

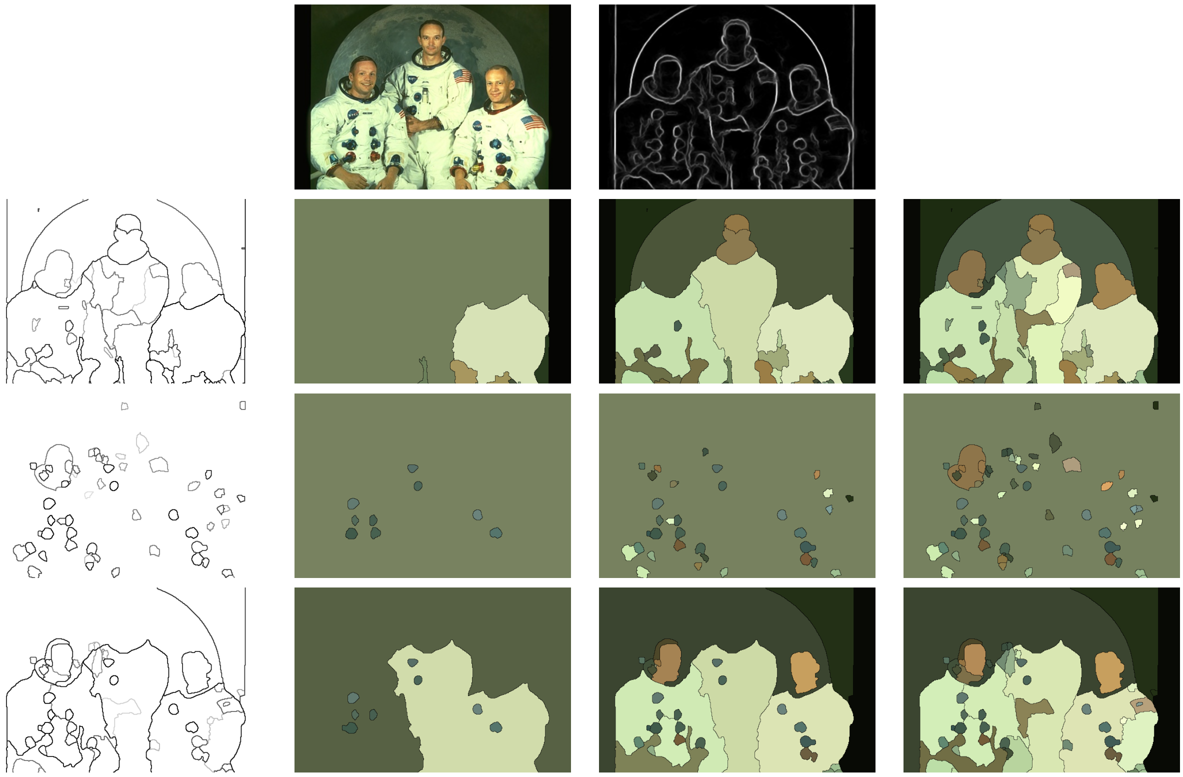


Figure 4.6: First line from left to right: original image I and the gradient G of I computed using the edge detector introduced in [29]. Second line from left to right: the saliency map of the hierarchical watershed $\mathcal{H}_c$ of G based on regularized circularity attribute values and three partitions of $\mathcal{H}_c$ with 10, 35 and 60 regions, respectively. Third line from left to right: the saliency map of the circularity based hierarchy $\mathcal{H}_{cc}$, which is not a hierarchical watershed of G, and three partitions of $\mathcal{H}_{cc}$ with 10, 35 and 60 regions, respectively. Fourth line: the watersheding $\mathcal{H}_w$ of $\mathcal{H}_{cc}$ and three partitions of $\mathcal{H}_w$ with 10, 35 and 60 regions, respectively.

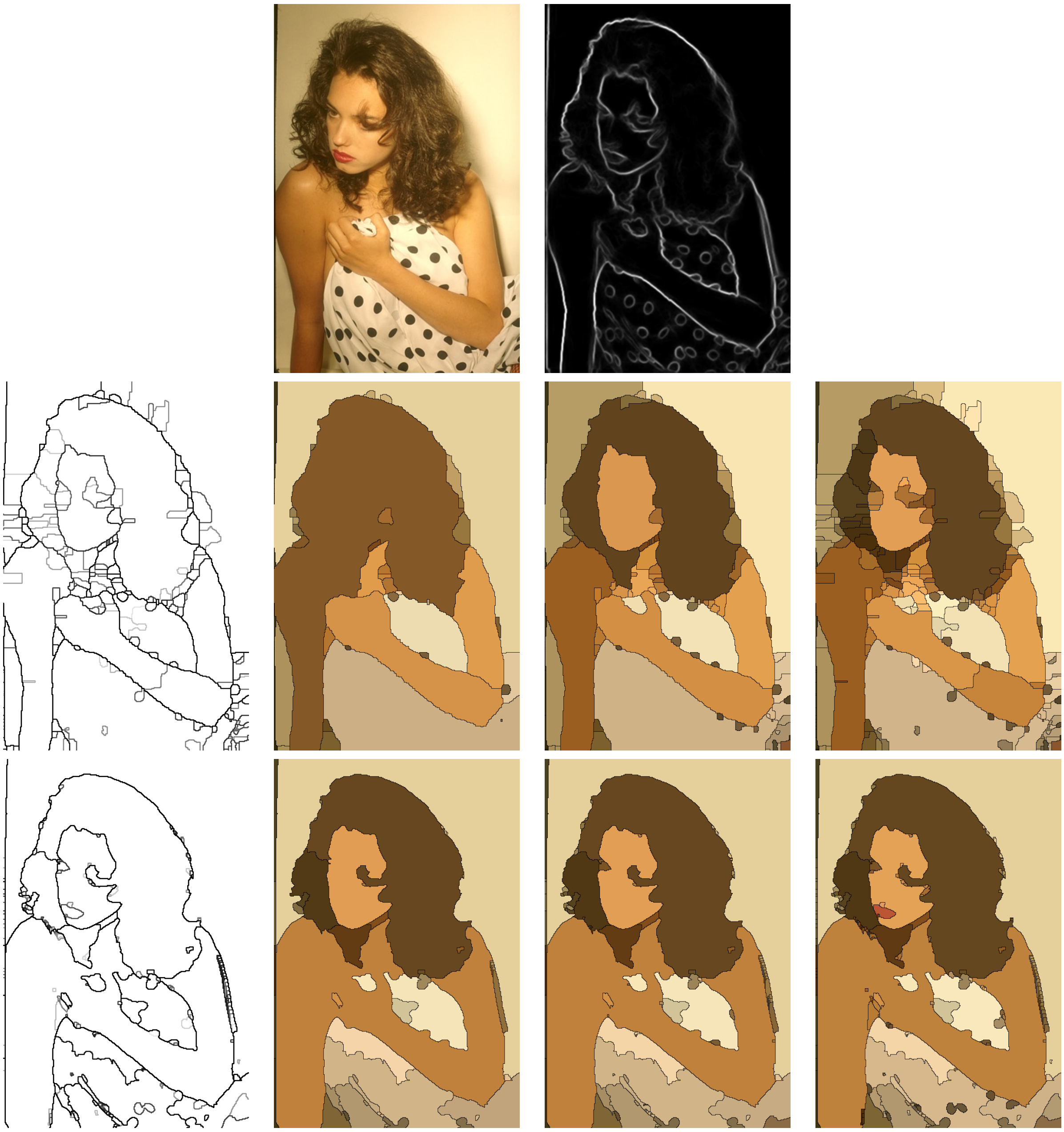


Figure 4.7: First line from left to right: original image I and the gradient G of I computed using the edge detector introduced in [29]. Second line from left to right: the saliency map of the COB [60] hierarchy of I and three segmentations of $\mathcal{H}_{cob}$ with 50, 100 and 200 regions, respectively. Third line from left to right: the watersheding $\mathcal{H}_w$ of $\mathcal{H}_{cob}$ and three segmentations of $\mathcal{H}_w$ with 50, 100 and 200 regions, respectively.

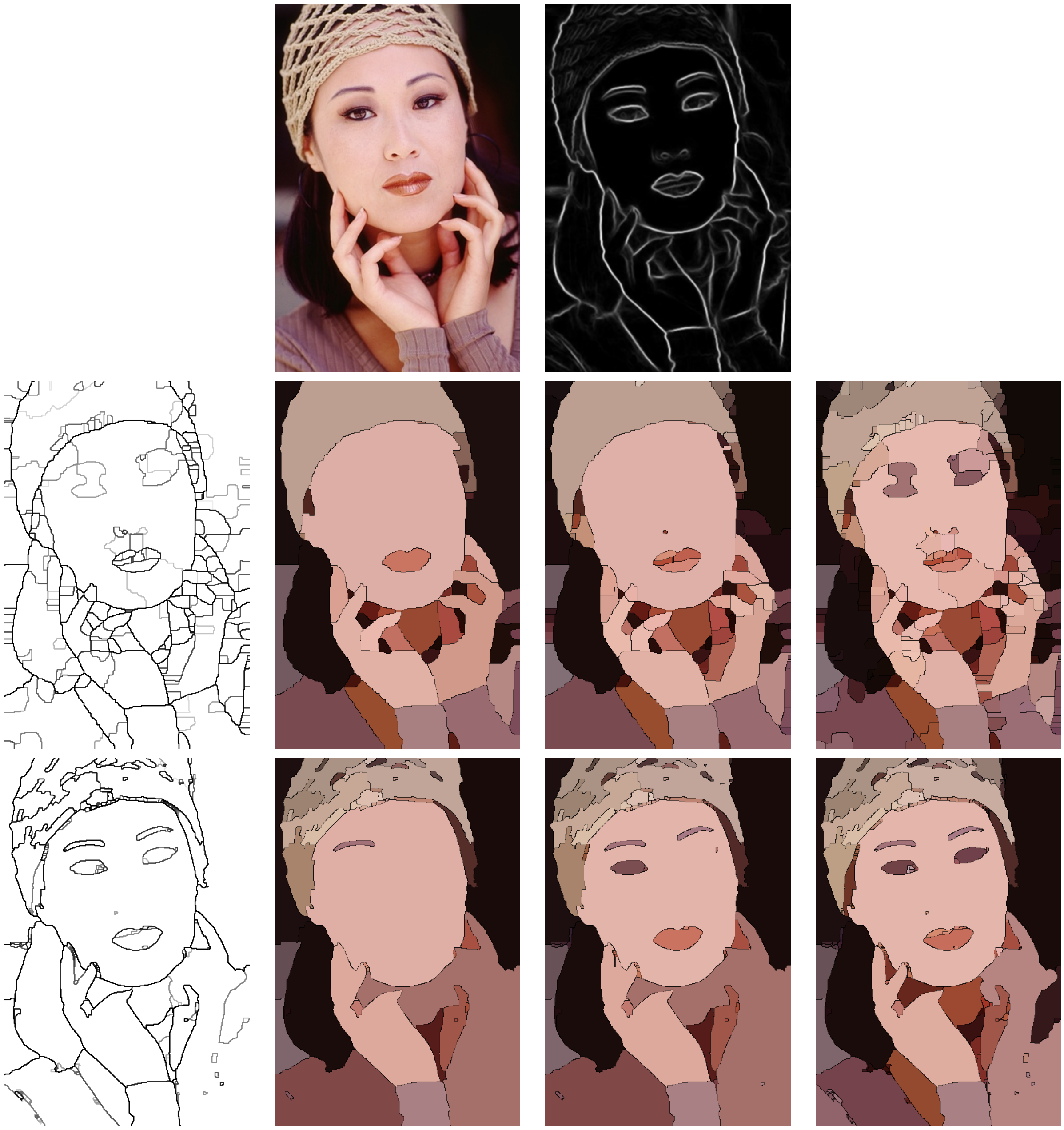


Figure 4.8: First line from left to right: original image I and the gradient G of I computed using the edge detector introduced in [29]. Second line from left to right: the saliency map of the COB [60] hierarchy of I and three segmentations of $\mathcal{H}_{cob}$ with 50, 100 and 200 regions, respectively. Third line from left to right: the watersheding $\mathcal{H}_w$ of $\mathcal{H}_{cob}$ and three segmentations of $\mathcal{H}_w$ with 50, 100 and 200 regions, respectively.

Chapter 5

Probability of hierarchical watersheds

In this chapter, we introduce the notion of probability of a hierarchical watershed and an efficient algorithm to compute such probability. This chapter comprises the results of the following article:

- D. S. Maia, J. Cousty, L. Najman, and B. Perret. On the probabilities of hierarchical watersheds. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 137–149. Springer, 2019.

5.1 Introduction

In the context of digital image processing, gray-level images can be treated as topographic surfaces whose altitudes are determined by the pixel gray-levels. The local minima of an image are the regions of uniform grey-level surrounded by pixels of strictly higher gray-levels. We show the representation of a gray-scale image with four local minima and a watershed segmentation in Figure 5.1(a) and (b), respectively.

Hierarchical watersheds are sequences of nested segmentations equivalent to filterings of an initial watershed segmentation. Let I be an image. The construction of a hierarchical watershed of I is often based on a criterion used to order the minima of I, as the area and the dynamics [98, 45]. More specifically, given any total ordering $\prec$ on the set of minima of I, the hierarchical watershed of I for $\prec$ is constructed by iteratively "flooding" the minima of I according to $\prec$. For instance, let us consider the total ordering $\prec$ on the set of minima $\{A, B, C, D\}$ of the image I of Figure 5.1(a) such that $C \prec D \prec B \prec A$. In Figure 5.1(b), (c), (d) and (e), we show the sequence of floodings of the minima of I for $\prec$.

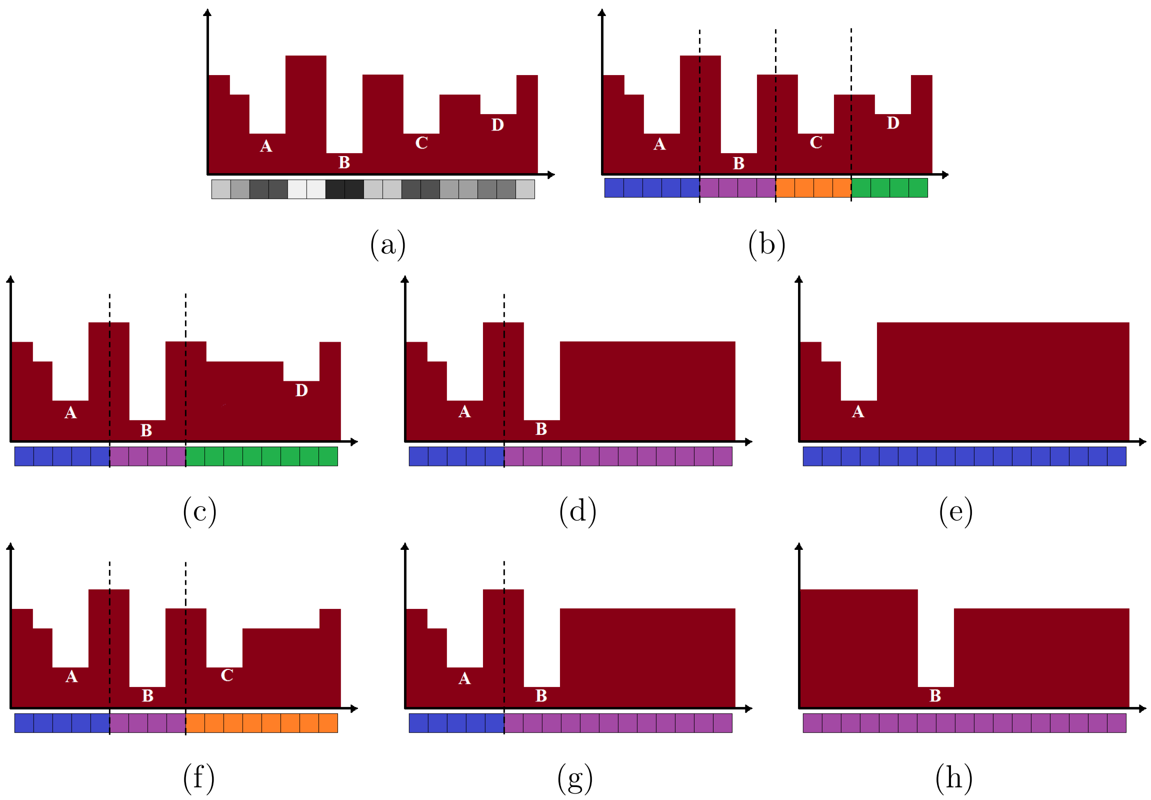


Figure 5.1: (a): A gray-scale image I with four minima. (b) A watershed segmentation of I: the vertical dashed lines represent the watershed-cuts. (c), (d) and (e): The watershed segmentations resulting from iteratively flooding the minima C, D and B, respectively. (f), (g) and (h): The watershed segmentations resulting from iteratively flooding the minima D, C and A, respectively.

The watershed segmentation of those floodings compose the hierarchical watershed of I for $\prec$.

In fact, we may obtain the same hierarchical watershed for several total orderings on the set of minima of an image. For example, we show in Figure 5.1(b), (f), (g) and (h) the floodings of the minima of the image I for another total ordering $\prec'$ such that $D \prec' C \prec' A \prec' B$. We can observe that the floodings for the total orderings $\prec'$ and $\prec$ induce the same sequence of watershed segmentations. Indeed, given any image I and any hierarchical watershed $\mathcal{H}$ of I, there may exist several total orderings on the set of minima of I whose hierarchical watersheds correspond to $\mathcal{H}$. In other words, it is possible to order the minima of I according to distinct criteria and still obtain the same hierarchical watershed.

In this study, (the gradients of) images are represented as weighted graphs. We define the probability of a hierarchical watershed $\mathcal{H}$ as the probability of $\mathcal{H}$ to be the hierarchical watershed of a given weighted graph (G, w) for an arbitrary sequence of minima of w. Let (G, w) be a weighted graph and let $\mathcal{H}$ be a hierarchical watershed of (G, w). In this study, we tackle the following problems:

(P_1) Find the probability of $\mathcal{H}$ to be the hierarchical watershed of (G, w) for an arbitrary sequence of minima of w;

(P_2) Characterize the most probable hierarchical watersheds of (G, w); and

(P_3) Characterize the least probable hierarchical watersheds of (G, w).

Other studies related to probability and (watershed) segmentations are found in [2, 9, 95, 51]. In [2], a stochastic watershed segmentation based on random markers is introduced. In [9], the definitions of watersheds with multiple solutions for a single image are unified in the definition of tie-zone watersheds, which returns a unique solution. In [95], the authors propose a method to list the k-minimum spanning trees that induce distinct segmentations for a given set of markers. In [51], the authors estimate the probability that any two regions of a watershed segmentation have the same texture, which is further used to build hierarchies of segmentations.

In Section 5.2, we solve problem (P_1) for the case where the given graph (G, w) has pairwise distinct edge weights. The solution to (P_1) is based on the bijection between hierarchies of partitions and saliency maps (see Section 2.4.1) and on the link between hierarchical watersheds and binary partition hierarchies (see Section 2.6.1). Then, we propose a quasi-linear time algorithm to compute the probability of a hierarchical watershed. In Section 5.4, we characterize the most and least probable hierarchical watersheds of (G, w) and we provide an algorithm to obtain such hierarchies.

Important notations: in the remainder of this chapter, the symbol G denotes a tree. To shorten the notation, the vertex set of G is denoted by V and its edge set is denoted by E. The symbol w denotes a map from E into $\mathbb{R}$ such that, for any pair of distinct edges u and v in E, we have $w(u) \neq w(v)$. Thus, the pair (G, w) is a weighted graph. Every hierarchy considered in this chapter is connected for G and therefore, for the sake of simplicity, we use the term *hierarchy* instead of *hierarchy which is connected for G*. We denote by n the number of minima of w. Every sequence of minima of w considered in this chapter is a sequence of n pairwise distinct minima of w and, therefore, we use the term *sequence of minima of w* instead of *sequence of n pairwise distinct minima of w*. The set of all sequences of minima of w is denoted by $\mathcal{M}_w$. By abuse of terminology, when no confusion is possible, if M is a minimum of w, we call the set $V(M)$ of vertices of M as a minimum of w.

5.2 Studying probabilities of hierarchical watersheds

Let $\mathcal{H}$ be a hierarchical watershed of (G, w). Then, by the definition of hierarchical watersheds (see Definition 1), there is a sequence $\mathcal{S}$ of minima of w such that $\mathcal{H}$ is the hierarchical watershed of (G, w) for $\mathcal{S}$. Indeed, as illustrated by the graphical example of Figure 5.1, the hierarchy $\mathcal{H}$ may be the hierarchical watershed of (G, w) for several sequences of minima of w. The *equivalence class of sequences of minima for* $\mathcal{H}$, denoted by $S_w(\mathcal{H})$, is the set which contains every sequence S of minima of w such that $\mathcal{H}$ is the hierarchical watershed of (G, w) for $\mathcal{S}$. We formalize the notion of probability of a hierarchical watershed in the following definition.

Definition 34 (probability of a hierarchical watershed). *Let $\mathcal{H}$ be a hierarchical watershed of (G, w). Let $\mathcal{S}$ be uniformely distributed on the set $\mathcal{M}_w$ of all sequences of minima of w. We define* the probability of $\mathcal{H}$ knowing w, *denoted by $p(\mathcal{H}|w)$, as the probability that the hierarchical watershed of (G,w) for $\mathcal{S}$ is equal to $\mathcal{H}$.*

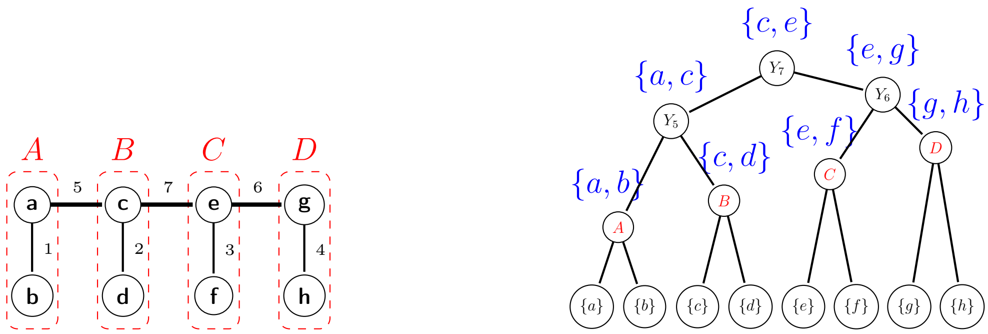
As established in the next property, the probability of a hierarchical watershed $\mathcal{H}$ is the ratio between the number of elements in the equivalence class of sequences of minima for $\mathcal{H}$ and the number of sequences of minima of w. To ease the reading of this chapter, the proof of some of the properties established here are presented in Appendix 8.3.

Property 35. *Let $\mathcal{H}$ be a hierarchical watershed of (G, w). The probability $p(\mathcal{H}|w)$ of $\mathcal{H}$ knowing w is the ratio k/n where k and n are the numbers of elements of $S_w(\mathcal{H})$ and of $\mathcal{M}_w$, respectively.*

In order to solve the problem of finding the probability of hierarchical watersheds, we first remind the definition of watershed-cut edges through an example, and then we introduce maximal regions.

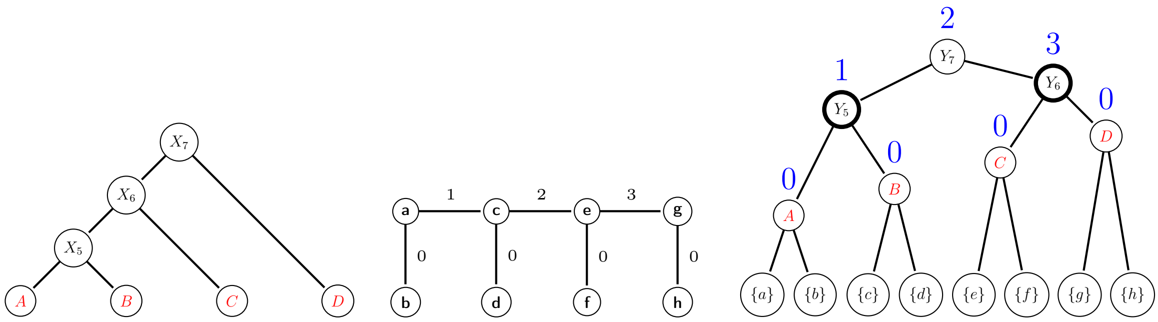
Important notations: as the edges of G have pairwise distinct weights for w, there is only one altitude ordering for w. We denote by $\prec$ the unique altitude ordering for w and we denote by $\mathcal{B}_\prec$ the binary partition hierarchy of (G, w) by $\prec$. Given an edge u in E, we denote by R_u the region of $\mathcal{B}_\prec$ whose building edge is u. When no confusion is possible, any watershed-cut edge for $\prec$ (see Definition 2) will be simply called a watershed-cut edge.

Let (G, w) be the graph of Figure 5.2(a), let $\prec$ be the unique altitude ordering for w, and let $\mathcal{B}_\prec$ be the binary partition hierarchy by $\prec$ of Figure 5.2(b). We can observe that both children of each of the regions Y_5, Y_6 and Y_7 of $\mathcal{B}_\prec$ include at least one minimum of w. By the definition of watershed-cut edges for an altitude ordering (Definition 2), the building edges of the regions Y_5, Y_6 and Y_7 are the watershed-cut edges (for $\prec$).



(a) (G, w) (b) $\mathcal{B}_{\prec}$

Figure 5.2: (a) a weighted graph (G, w) with four minima delimited by the dashed rectangles. The watershed-cut edges for the unique altitude ordering $\prec$ for w are represented in bold. (b) the binary partition hierarchy $\mathcal{B}_{\prec}$ for the unique altitude ordering $\prec$.



(a) $\mathcal{H}$ (b) $\Phi(\mathcal{H})$ (c) $\mathcal{B}_{\prec}$ and $\Phi(\mathcal{H})$

Figure 5.3: (a): a hierarchical watershed of the graph (G, w) of Figure 5.2(a). (b): the saliency map $\Phi(\mathcal{H})$ of $\mathcal{H}$. (c): the saliency map $\Phi(\mathcal{H})$ represented on the binary partition hierarchy $\mathcal{B}_{\prec}$ of (G, w). The weight above each region of $\mathcal{B}_{\prec}$ is the weight of the building of this region for $\Phi(\mathcal{H})$. The maximal regions for $\Phi(\mathcal{H})$ are the regions Y_5 and Y_6 (in bold).

Definition 36 (maximal region)**.** *Let u be a watershed-cut edge, let f be a map from E into $\mathbb{R}$ and let u be an edge in E. We say that the region R_u* is a maximal region (of $\mathcal{B}_{\prec}$) for f *if the weight of u for f is greater than the weight of the building edge of any region included in R_u,* i.e., *if $f(u) > max\{f(v) \mid v \in E, R_v \subset R_u\}$.*

Let $\mathcal{H}$ be the hierarchy of Figure 5.3(a) and let (G, w) be the graph of Figure 5.2(a). We can verify that $\mathcal{H}$ is the hierarchical watershed of (G, w) for the sequences (A, B, C, D), (A, B, D, C), (B, A, C, D) and (B, A, D, C). The saliency map $\Phi(\mathcal{H})$ of $\mathcal{H}$ is represented in Figure 5.3(b). In Figure 5.3(c), the weight above each region R of $\mathcal{B}_{\prec}$ is the weight of the building edge of R for $\Phi(\mathcal{H})$. We can conclude that the only maximal regions for $\Phi(\mathcal{H})$ are the regions Y_5 and Y_6.

The following property, whose proof is detailed in Appendix 8.3.1, establishes that, given any hierarchical watershed $\mathcal{H}$ of (G, w), the probability of $\mathcal{H}$ knowing w can be defined through the number of maximal regions of $\mathcal{B}_{\prec}$ for the saliency map $\Phi(\mathcal{H})$ of $\mathcal{H}$.

Property 37. *Let $\mathcal{H}$ be a hierarchical watershed of (G, w) and let m be the number of maximal regions of $\mathcal{B}_{\prec}$ for the saliency map $\Phi(\mathcal{H})$ of $\mathcal{H}$. The probability of $\mathcal{H}$ knowing w is:*

$$p(\mathcal{H} \mid w) = \frac{2^m}{|\mathcal{M}_w|}. \tag{5.1}$$

For instance, let us consider the hierarchical watershed $\mathcal{H}$ of Figure 5.3(a). As stated previously, the hierarchy $\mathcal{B}_{\prec}$ has two maximal regions (Y_5 and Y_6) for $\Phi(\mathcal{H})$. Since the graph (G, w) of Figure 5.2(a) has four minima, there are 4! sequences of minima of w. By property 37, we conclude that the probability of $\mathcal{H}$ knowing w is $\frac{2^2}{4!}$. Indeed, as aforementioned, the hierarchy $\mathcal{H}$ is the hierarchical watershed of (G, w) for four sequences of minima of w: (A, B, C, D), (A, B, D, C), (B, A, C, D) and (B, A, D, C).

5.3 Algorithm to compute the probability of a hierarchical watershed

From Property 37, we derive a quasi-linear time algorithm (Algorithm 4) to compute the probability of a hierarchical watershed. The inputs of Algorithm 4 are a weighted graph $((V, E), w)$ with pairwise distinct edge-weights and the saliency map f of a hierarchical watershed of $((V, E), w)$. The first steps are to compute the unique altitude ordering $\prec$ for w and the binary partition hierarchy $\mathcal{B}$ by $\prec$. Then, at lines 3-4, we compute the minima of w, their number n and the watershed-cut edges using the algorithm proposed in [74]. Then, the **for** loop at lines 6-19 computes the number m of maximal regions of $\mathcal{B}$ for f. For each edge u in increasing order for $\prec$, we compute the maximal weight among the (building edges of the) regions included in R_u. If u is a watershed-cut edge and if $f(u)$ is strictly greater than the maximal weight among the regions strictly included in R_u, then u is a maximal region and m is incremented by one. Finally, the algorithm returns the probability of f knowing w: $\frac{2^m}{n!}$.

Let us now study the time complexity of Algorithm 4. Using the merge sort algorithm, the altitude ordering $\prec$ for w can be computed in time $O(|E| log|E|)$. As established in [74], the binary partition hierarchy $\mathcal{B}$ can be computed in quasi-linear time with respect to $|E|$. More precisely, the time complexity to compute the binary partition hierarchy $\mathcal{B}$ is $O(|E| \times \alpha(|V|))$, where α is a slowly growing inverse of the single-valued Ackermann function. Using the method proposed in [74], the watershed-cut edges and the minima of w can be obtained in linear time $O(|V|)$ through a single pass on the regions of the hierarchy $\mathcal{B}$. Regarding the computation of the number of maximal regions for f, each instruction of the **for** loop at lines 6-19 can be computed in constant time because each region of $\mathcal{B}$ has at most two children. Hence, the time complexity to compute the number of maximal regions for f is linear with respect to the number of building edges for $\prec$. Therefore, the overall time complexity of Algorithm 4 is $O(|E|log|E|)$.

Algorithm 4 Probability of hierarchical watersheds

Data: $((V, E), w)$: a weighted tree with ordered edges
f: the saliency map of a hierarchical watershed $\mathcal{H}$ of $((V, E), w)$
Result: the probability of $\mathcal{H}$ knowing w

1: Compute the unique altitude ordering $\prec$ for f $\triangleright$ $O(|E|log|E|)$
2: Compute the binary partition hierarchy $\mathcal{B}$ by $\prec$ $\triangleright$ $O(|E| \times \alpha(|V|))$ with [74]
3: Compute the minima of w and their number n $\triangleright$ $O(|V|)$ with [74]
4: Compute the watershed-cut edges for $\prec$ $\triangleright$ $O(|V|)$ with [74]
`// Computation of the number` m `of maximal regions of` $\mathcal{B}$ `for` f
5: $m \leftarrow 0$ $\triangleright$ $O(1)$
6: **for** each edge u in increasing order for $\prec$ **do** $\triangleright$ $O(|V|)$
7: $\varphi(u) \leftarrow f(u)$ $\triangleright$ $O(1)$
8: **for** each non-leaf child X of R_u **do** $\triangleright$ $O(1)$
9: $v \leftarrow$ the building edge of X $\triangleright$ $O(1)$
10: $\varphi(u) \leftarrow max(\varphi(u), \varphi(v))$ $\triangleright$ $O(1)$
11: **end for**
12: **if** u is a watershed-cut edge **then** hspace* $\triangleright$ $O(1)$
13: $v_1 \leftarrow$ the building edge of a child of R_u $\triangleright$ $O(1)$
14: $v_2 \leftarrow$ the building edge of the sibling of R_u $\triangleright$ $O(1)$
15: **if** $f(u) > \varphi(v_1)$ and $f(u) > \varphi(v_2)$ **then** $\triangleright$ $O(1)$
16: $m \leftarrow m + 1$ $\triangleright$ $O(1)$
17: **end if**
18: **end if**
19: **end for**
return $\frac{2^m}{n!}$

5.4 Most and least probable hierarchical watersheds

In this section, we establish the upper and lower bounds on the probability of a hierarchical watershed, and a characterization of the most and least probable hierarchical watersheds of (G, w).

Let ℓ be the number of watershed-cut edges for $\prec$. By Definition 36, we can affirm that there are at most ℓ maximal regions of $\mathcal{B}_{\prec}$ for the saliency map of any hierarchical watershed of (G, w). Thus, we can derive the following Corollary 38, which establishes the tight upper bound on the probability of any hierarchical watershed of (G, w).

Corollary 38. *Let ℓ be the number of watershed-cut edges for $\prec$ and let $\mathcal{H}$ be a hierarchical watershed of (G, w). The tight upper bound on the probability of $\mathcal{H}$ knowing w is $\frac{2^\ell}{|\mathcal{M}_w|}$.*

Let u be a watershed-cut edge. We say that R_u *is a primary region of* $\mathcal{B}_{\prec}$ if there is no watershed-cut edge v such that $R_v \subset R_u$. Let f be the saliency map of a hierarchical watershed of (G, w). One can note that the value of f is zero (resp. non-zero) on non watershed-cut edges (resp. watershed-cut edges). Then, only the watershed-cut edges have non-zero weights for f. Let u be a watershed-cut edge. If R_u is a primary region of $\mathcal{B}_{\prec}$, then $f(u)$, being non-zero, is greater than $max\{f(v), v \in E \mid R_v \subset R_u\}$, which is equal to zero. Consequently, the region R_u is a maximal region of $\mathcal{B}_{\prec}$ for f. We conclude that each primary region of $\mathcal{B}_{\prec}$ is a maximal region of $\mathcal{B}_{\prec}$ for the saliency map of any hierarchical watershed of (G, w). We can now define the tight lower bound on the probability of a hierarchical watershed.

Corollary 39. *Let k be the number of primary regions of $\mathcal{B}_{\prec}$ and let $\mathcal{H}$ be a hierarchical watershed of (G, w). The tight lower bound on the probability of $\mathcal{H}$ knowing w is $\frac{2^k}{|\mathcal{M}_w|}$.*

If the map w has more than two minima, then there is at least one watershed-cut edge u such that R_u is not a primary region of $\mathcal{B}_{\prec}$. Therefore, the tight lower bound and the tight upper bound on the probabilities of hierarchical watersheds of (G, w) are not equal. This justifies the following definition of most probable hierarchical watersheds.

Definition 40 (most probable hierarchical watersheds)**.** *Let $\mathcal{H}$ be a hierarchical watershed of (G, w). We say that $\mathcal{H}$ is a* most probable hierarchical watershed of (G, w)*, if, for any hierarchical watershed $\mathcal{H}'$ for (G, w), we have $p(\mathcal{H} \mid w) \geq p(\mathcal{H}' \mid w)$.*

Let f be the saliency map of a hierarchical watershed of (G, w). Let u be an edge in E. By abuse of notation, we define the *weight of R_u for f* as the weight $f(u)$ of u. Let $\mathcal{H}$ be a hierarchical watershed of (G, w). By Corollary 38, the probability of $\mathcal{H}$ knowing w is maximal when, for every watershed-cut edge u, the region R_u is a maximal region of $\mathcal{B}_{\prec}$ for $\Phi(\mathcal{H})$. By the definition of maximal regions, we can establish the following characterization of the most probable hierarchical watersheds of (G, w).

Corollary 41. *Let $\mathcal{H}$ be a hierarchical watershed of (G, w). The hierarchy $\mathcal{H}$ is a most probable hierarchical watershed of (G, w) if and only if the weights for $\Phi(\mathcal{H})$ are increasing on the regions of the hierarchy $\mathcal{B}_{\prec}$.*

Let $\mathcal{H}$ be a most probable hierarchical watershed of (G, w). By Corollary 41, we may conclude that the order in which the regions of $\mathcal{H}$ are merged along the partitions of $\mathcal{H}$ are constrained by the hierarchy $\mathcal{B}_{\prec}$. Thus, we can deduce the following Corollary from Property 41.

Corollary 42. *Let $\mathcal{H}$ be a hierarchical watershed of (G, w). The hierarchy $\mathcal{H}$ is a most probable hierarchical watershed of (G, w) if and only if each non-leaf region of $\mathcal{H}$ is a region of $\mathcal{B}_{\prec}$.*

Let (G, w) be the weighted graph of Figure 5.2(a). In Figure 5.4, we present four hierarchical watersheds of (G, w). Indeed, those are the only hierarchical watersheds of (G, w). For each hierarchy, we show its saliency map represented on the graph G and on the binary partition hierarchy $\mathcal{B}_{\prec}$. The dominant regions for each saliency map are in bold. Since w has four minima, the number of sequences of minima of w is $4!$. The probability of the hierarchies $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathcal{H}_3$ and $\mathcal{H}_4$ knowing w are $\frac{4}{4!}, \frac{4}{4!}, \frac{8}{4!}$ and $\frac{8}{4!}$, respectively. Therefore, the set of most probable hierarchical watersheds of (G, w) is $\{\mathcal{H}_3, \mathcal{H}_4\}$. We can verify that each non-leaf region of the hierarchies $\mathcal{H}_3$ and $\mathcal{H}_4$ is a region of $\mathcal{B}_{\prec}$, which is not the case for $\mathcal{H}_1$ and $\mathcal{H}_2$. This example illustrates Corollary 42.

Following the same idea of the definition of most probable hierarchical watersheds, we introduce in the next definition the notion of least probable hierarchical watersheds.

Definition 43 (least probable hierarchical watersheds)**.** *Let $\mathcal{H}$ be a hierarchical watershed of (G, w). We say that $\mathcal{H}$ is a* least probable hierarchical watershed of (G, w)*, if, for any hierarchical watershed $\mathcal{H}'$ for (G, w), we have $p(\mathcal{H} \mid w) \leq p(\mathcal{H}' \mid w)$.*

Let $\mathcal{H}$ be a least probable hierarchical watershed of (G, w). By Corollary 39, only the primary regions of $\mathcal{B}_{\prec}$ are maximal regions for the saliency map $\Phi(\mathcal{H})$. Hence, for any watershed-cut edge u, if R_u is not a primary region of $\mathcal{B}_{\prec}$, then there is at least one watershed-cut edge v such that $R_v \subset R_u$ and such that $\Phi(\mathcal{H})(v) \geq \Phi(\mathcal{H})(u)$. Moreover, by the characterization of hierarchical watersheds provided in Chapter 3 (Lemma 19), which link the notions of hierarchical watersheds and one-side increasing maps (see Definition 18), we can affirm that $\Phi(\mathcal{H})$ is one-side increasing for $\prec$. The later implies that, for any edge u, there exists at least one child X of R_u such that $\Phi(\mathcal{H})(u)$ is greater than the weight (of the building edges) of all regions included in X. Hence, we can deduce the following characterization of least probable hierarchical watersheds.

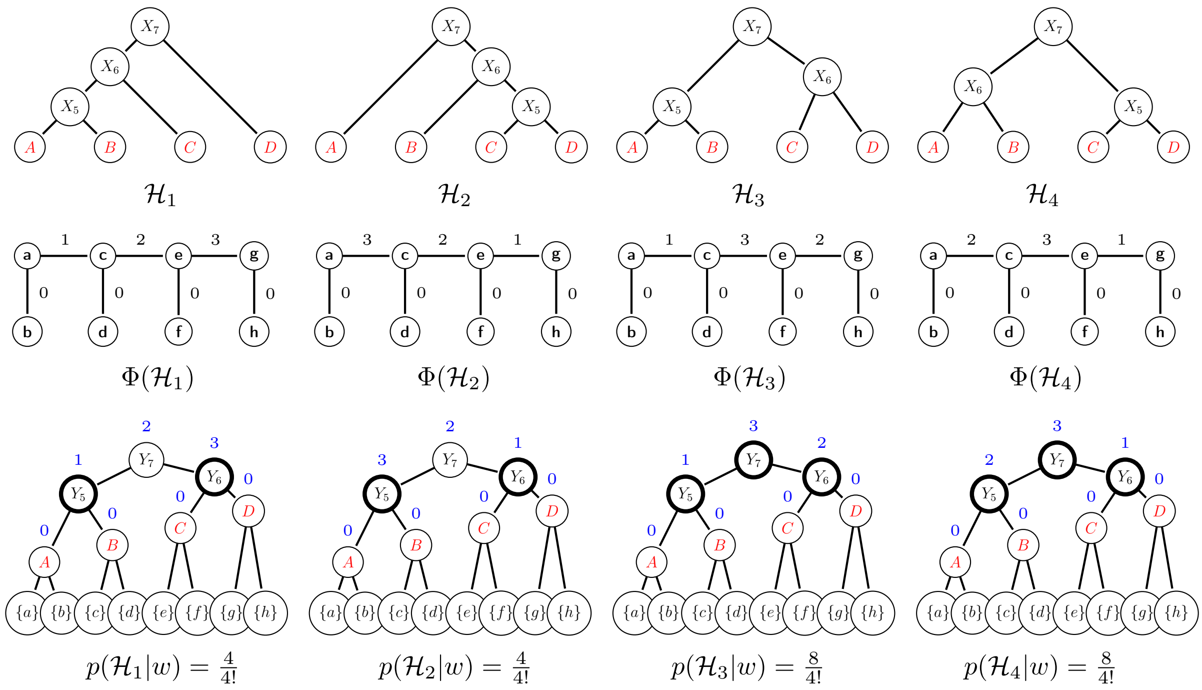


Figure 5.4: The hierarchical watersheds for the weighted graph (G, w) of Figure 5.2(a), their saliency maps represented on the graph (G, w) and on the binary partition hierarchy $\mathcal{B}$. The dominant regions of $\mathcal{B}_\prec$ for each saliency map is represented in bold. The probability of each hierarchical watershed knowing w is presented under $\mathcal{B}$.

Corollary 44. *Let $\mathcal{H}$ be a hierarchical watershed of (G, w). The hierarchy $\mathcal{H}$ is a least probable hierarchical watershed of (G, w) if and only if, for any watershed-cut edge u such that R_u is not a primary region of $\mathcal{B}_\prec$ for $\Phi(\mathcal{H})$, there is exactly one child X of R_u such that $\Phi(\mathcal{H}) \geq max\{\Phi(\mathcal{H})(v) \mid R_v \subseteq X\}$.*

Let (G, w) be the weighted graph of Figure 5.2(a) and let $\mathcal{B}_\prec$ be the unique binary partition hierarchy of (G, w) shown in Figure 5.2(c). We can observe that the regions Y_5 and Y_6 of $\mathcal{B}_\prec$ are primary regions, by Y_7 is not. Hence, the building edge $u = \{c, e\}$ of X_7 is the unique watershed-cut edge such that R_u is not a primary region. Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be the hierarchical watersheds of (G, w) shown in Figure 5.4(a) and (b), respectively. We can observe that $\Phi(\mathcal{H}_1)(u)$ (resp. $\Phi(\mathcal{H}_2)(u)$) is only greater than the weights of the regions included in the child Y_5 (resp. Y_6). By Corollary 44, we conclude that the set of least probable hierarchical watersheds of (G, w) is $\{\mathcal{H}_1, \mathcal{H}_2\}$.

5.5 Algorithms to compute a most and a least probable hierarchical watershed

From Corollary 41, we deduce a recursive algorithm to find the saliency map f of a most probable hierarchical watershed of (G, w). Let ℓ be the number of watershed-cut edges for $\prec$ and let L be the set $\{1, \ldots, \ell\}$. Let u be the building edge of the region V of $\mathcal{B}_{\prec}$. First, we assign $f(u)$ to $max\{1, \ldots, \ell\}$. Let R' and R'' be the children of V and let n' and n'' be the number of minima of w included in R' and R'', respectively. Subsequently, we arbitrarily divide the set $\{1, \ldots, \ell - 1\}$ into two subsets L' and L'' with $n' - 1$ and $n'' - 1$ elements, respectively. Then, the sets L' and L'' are propagated to R' and R'', respectively. The subtrees rooted in R' and R'' are treated separately. This process is performed until the weights of all edges are assigned.

Algorithm 5 describes our method to obtain an arbitrary most probable hierarchical watershed of a graph. The input is a graph $((V, E), w)$ which is a tree with pairwise distinct edge weights. The first step is to order the edges in E to obtain the unique altitude ordering $\prec$ for w. Then, we compute the binary partition hierarchy by $\prec$. Subsequently, for any edge u in E, the **for** loop at lines 3-8 computes the number $M(u)$ of minima included in R_u, and assigns $ws(u)$ to 1 if u is a watershed-cut edge for $\prec$. At lines 9-14, we declare a linked list L and insert all integers from 1 to the number of minima of w. Then, the values of the map f are computed by the recursive function SaliencyMostProbable. In the function SaliencyMostProbable, for each edge u, if u is a watershed-cut edge for $\prec$, then $f(u)$ is assigned to the maximal value in the list L and, otherwise, it is assigned to zero. Then, this function is called for each non-leaf child X of R_u by passing in parameter another linked list LL with $M(v) - 1$ random elements of L, where v is the building edge of X.

Let us now analyse the complexity of Algorithm 5. Since the input graph is a tree, the number of edges in E is the number of vertices in V plus one. Hence, the altitude ordering $\prec$ can be obtained in time $O(|V|log|V|)$ by using the merge sort algorithm to order the edges in E. The overall time complexity to compute the the binary partition hierarchy $\mathcal{B}$ by $\prec$, the minima of w and the watershed-cuts edges for $\prec$ is quasi-linear with respect to $|E|$ using the algorithm proposed in [74]. As the number of minima of w is at most $\frac{|V|}{2}$, the **for** loop at lines 11-14 is executed in linear time with respect to $|V|$. In order to define the weights of the map f, the recursive function SaliencyMostProbable is called once for every edge in E, starting from the building edge of V. For each call, a number k of elements in L are randomly chosen and added to a new linked list LL, where k is the number of minima included in a region of $\mathcal{B}$ minus one. In the worst case, the value k is equal to $\frac{|V|}{2}$, so we can consider that the time complexity to execute the **for** loop of the function SALIENCYMOSTPROBABLE is linear with respect to $|V|$. As the function SALIENCYMOSTPROBABLE is called $|E| = |V| + 1$ times, we can say that the time complexity to obtain the values of the map f is quadratic with respect to $|V|$ in the worst case. Therefore, the time complexity to compute the saliency map of a most probable hierarchical watershed with Algorithm 5 is $O(|V|^2)$.

Algorithm 5 Most probable hierarchical watershed

Data: $((V, E), w)$: a weighted graph (tree) with pairwise distinct edge weights
Result: the saliency map f of a most probable hierarchical watershed of $((V, E), w)$

Compute the altitude ordering $\prec$ for w $\triangleright$ $O(|V|log|V|)$
Compute the binary partition hierarchy $\mathcal{B}$ by $\prec$ $\triangleright$ $O(|V| \times \alpha(|V|))$ with [74]
for each edge u in E in increasing order for $\prec$ **do** $\triangleright$ $O(|V|)$ with [74]
    $\mu(u) \leftarrow$ number of minima included in R_u
    if u is a watershed-cut edge for $\prec$ **then**
        $ws(u) \leftarrow 1$
    end if
end for
Declare L as a linked list of integers $\triangleright$ $O(1)$
$i \leftarrow 1$ $\triangleright$ $O(1)$
while $i <$ number of minima of w **do** $\triangleright$ $O(|V|)$
    add i to L $\triangleright$ $O(1)$
    $i \leftarrow i + 1$ $\triangleright$ $O(1)$
end while
$u \leftarrow$ building edge of the region V of $\mathcal{B}$ $\triangleright$ $O(1)$
SALIENCYMOSTPROBABLE(u, f, $\mathcal{B}$, ws, μ, L) $\triangleright$ $O(|V|^2)$
    return f

From Corollary 44, we designed a recursive algorithm to compute the saliency map of a least probable hierarchical watershed. The idea is similar to our algorithm to compute the saliency map of a most probable hierarchical watersheds. A list of integer weights is propagated from the root of the binary partition hierarchy (of the input graph) to the leaf regions. To each watershed-cut edge u, we assign a weight in the list and, then, the remaining integers are split and propagate to the children of R_u. The split is performed in a way that exactly one child of R_u receives a list of weights that are all smaller than the weight assigned to u. This method guarantees that the final map is one-side increasing (see Definition 18) and, hence, that it is saliency map of a hierarchical watershed by Lemma 19.

Function 6 SALIENCYMOSTPROBABLE(u, f, $\mathcal{B}$, ws, μ, L)

if $ws(u) == 1$ **then** ▷ $O(1)$
  $f(u) \leftarrow$ maximum integer in the list L ▷ $O(1)$
  delete the value $f(u)$ from the list L ▷ $O(1)$
else
  $f(u) \leftarrow 0$ ▷ $O(1)$
end if
for each non-leaf child X of R_u **do** ▷ $O(1)$
  $v \leftarrow$ building edge of X ▷ $O(1)$
  declare LL as a linked list of integers ▷ $O(1)$
  insert $\mu(v) - 1$ distinct elements of L in LL and delete them from L ▷ $O(|V|)$
  SALIENCYMOSTPROBABLE(v, f, $\mathcal{B}$, ws, μ, LL) ▷ $O(1)$
end for

Our algorithm to obtain a least probable hierarchical watershed of a weighted graph is detailed in Algorithm 7. The input is a weighted graph $((V, E), w)$ which is a tree with pairwise distinct edge weights. The output is the saliency map f of a least probable hierarchical watershed of $((V, E), w)$. The altitude ordering $\prec$ for w, the binary partition hierarchy $\mathcal{B}$ by $\prec$, the minima of w, the watershed-cut edges for $\prec$, and the linked list L are obtained in the first fifteen lines of Algorithm 7, which are equivalent to the same lines of Algorithm 5. Then, the map f is computed by the recursive function SALIENCYLEASTPROBABLE. In this function, for each watershed-cut edge u, $f(u)$ is assigned to a value k in the list L such that k is not maximal. Then, a list of values in L which are smaller than $f(u)$ is propagated to a child X of R_u. The remaining values in L are propagated to the sibling of X. In the case where u is not a watershed-cut edge, $f(u)$ is assigned to 0 and the list L is propagated to a non-leaf region of R_u, if any.

We can observe that the linked lists passed in parameter remain ordered for every call of the function SALIENCYMOSTPROBABLE. Hence, like the function SALIENCYMOSTPROBABLE, the time complexity to execute any instruction of the function SALIENCYLEASTPROBABLE is at most linear with respect to $|V|$. Since the function SALIENCYLEASTPROBABLE is called $|E| = |V| - 1$ times, the overall time complexity of Algorithm 7 is $O(|V|^2)$.

Illustrations of Algorithms 5 and 7 are presented in Figure 5.5. The underlying graph of each gradient G_1 and G_2 illustrated in Figure 5.5 does not have pairwise distinct edge weights. This implies that there may be several altitude orderings and, consequently, several binary partition hierarchies of each of those graphs. Hence, we worked with a fixed altitude ordering $\prec$ for each graph. In Figure 5.5 we show two of the most and two of the least probable hierarchical watersheds of each image gradient (for a fixed altitude ordering).

Algorithm 7 Least probable hierarchical watershed

```
Data: ((V, E), w): a weighted graph (tree) with pairwise distinct edge weights
Result: the saliency map f of a least probable hierarchical watershed of ((V, E), w)
/* Lines 1 − 15 of Algorithm 5 */
SaliencyLeastProbable(u, f, B, ws, μ, L)                              ▷ O(|V|^2)
```

Function 8 SALIENCYLEASTPROBABLE(u, f, $\mathcal{B}$, ws, μ, L)

```
if ws(u) == 1 then                                                     ▷ O(1)
    v1 ← building edge of a child X of R_u                             ▷ O(1)
    v2 ← building edge of the sibling of X                             ▷ O(1)
    k ← random value between min(μ(v1), μ(v2)) and μ(u) − 2            ▷ O(1)
    f(u) ← L(k)                                                        ▷ O(1)
    delete the value L(k) from the list L                              ▷ O(1)
    declare LL as a linked list of integers                            ▷ O(1)
    i ← 1                                                              ▷ O(1)
    while i < μ(v1) do                                                 ▷ O(|V|)
        insert L(i) in LL
        delete the integer L(i) from L
    end while
    SaliencyLeastProbable(v1, f, B, ws, μ, LL)                         ▷ O(1)
    SaliencyLeastProbable(v2, f, B, ws, μ, L)                          ▷ O(1)
else
    f(u) ← 0                                                           ▷ O(1)
    if R_u has a non-leaf child then                                   ▷ O(1)
        v ← building edge of the non-leaf child of R_u                 ▷ O(1)
        SaliencyLeastProbable(v, f, B, ws, μ, L)                       ▷ O(1)
    end if
end if
```

5.6 Discussion and conclusion

In this chapter, we introduced the notion of probability of a hierarchical watershed computed from a sequence of minima of a weighted graph. Then, we presented an efficient method to obtain the probability of any hierarchical watershed. We also introduced the notions of most and least probable hierarchical watersheds and algorithms to obtain such hierarchies.

A practical application of this study is the analysis of hierarchical watersheds based on increasing criteria. Given a list $A_1, \ldots, A_\ell$ of increasing criteria, is there a criterion A_i such that the hierarchical watersheds based on A_i are more probable than the hierarchical watersheds based on the other criteria of this list? For instance, are dynamic-based hierarchical watersheds more probable than area-based hierarchical watersheds?

Now, let us consider the set S_m (resp. S_l) of sequences of minima of w such that, for any sequence $\mathcal{S}$ in S_m (resp. S_l), the hierarchical watershed for $\mathcal{S}$ is one of the most (resp. least) probable hierarchical watersheds of (G, w). Another relevant question related to this topic is: are the sequences of minima ordered according to the usual increasing criteria (e.g. area and volume) more likely to fall into the set S_m than into the set S_l? In other words, based on the usual increasing criteria, is it more likely to obtain a most or a least probable hierarchical watershed?

Finally, it would be interesting to evaluate hierarchical watersheds with respect to their probabilities in order to investigate the link between the probability of a hierarchy and its performance.

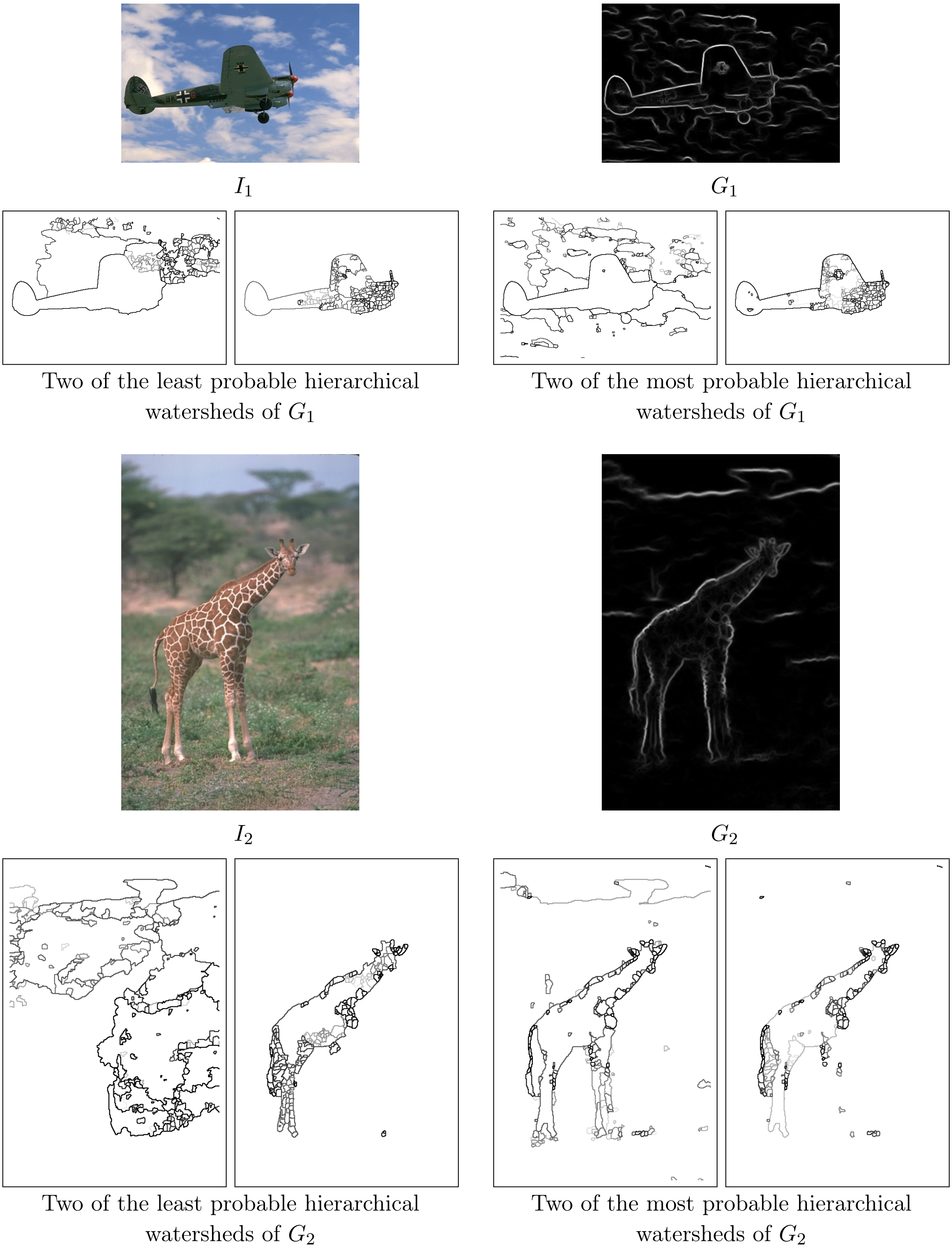


Figure 5.5: Two images I_1 and I_2, their gradients G_1 and G_2 computed using the edge detector proposed in [29], two of the least probable hierarchical watersheds of G_1 and G_2, and two of the most probable hierarchical watersheds of G_1 and G_2.

Chapter 6

Evaluation framework of hierarchies of segmentations

In this chapter, we present the evaluation framework of hierarchies of segmentations introduced in the following article:

- B. Perret, J. Cousty, S. J. F. Guimaraes, and D. S. Maia. Evaluation of hierarchical watersheds. *IEEE Transactions on Image Processing*, 27(4):1676–1688, 2017.

This framework will be further used in the evaluation of combinations of hierarchies studied in Chapter 7.

6.1 Introduction

In the past few decades, several hierarchical segmentation methods have been introduced for different purposes, including image simplification/filtering [13, 89], scene labeling [32], (shape based) object detection [105, 38], and object tracking in videos [46]. The vast range of hierarchical segmentation methods and their diversified applications call for an evaluation framework that covers multiple aspects of each method.

In the context of image segmentation, the most commonly used evaluation frameworks are based on empirical assessment of segmentations. Given an image I and an image segmentation algorithm A, we measure the quality of A by measuring the dissimilarity between the segmentation of I (computed with A) and a ground-truth segmentation of I. The numerous dissimilarity measures studied in the literature can be classified into boundary-based and region-based measures. Boundary-based measures evaluate how much the boundaries/contours between regions of a segmentation match the ground-truth contours. In turn, region-based measures take into account the matching between the regions of a segmentation and the regions of a ground-truth segmentation.

Methods to evaluate hierarchies of segmentations arise naturally by combining the individual evaluation of the partitions of a hierarchy. Given a hierarchy $\mathcal{H}$ and a measure of quality M, the evaluation of $\mathcal{H}$ with respect to M can be given as a combination, *e.g.* average or median, of the evaluation of all partitions of $\mathcal{H}$ against a given image ground-truth. For instance, this is the spirit of the evaluation framework proposed in [5].

A different evaluation approach is proposed in [97]. To optimize parameters related to color and texture in the computation of hierarchies of partitions of remote sensing data, [97] proposes an evaluation framework with takes into consideration the importance of a contour/edge in the ground-truth segmentation. More precisely, ground-truth edges are classified into compulsory and optional. Their assessment is based on verifying the percentage of ground-truth edges that correspond to missed or false detection. A missed compulsory edge weighs more in the final assessment than a missed optional edge. Moreover, the lower is the level where a compulsory edge vanishes in the hierarchy, the greater is the error. Conversely, the greater is the level that a false detected edge is present in the hierarchy, the greater is the error. [97] successfully incorporates hierarchical contours information into their assessment, though, it requires ground-truth edges to be pre-classified by their importance, which is not the case for most image datasets.

In this chapter, we present the evaluation framework of hierarchies introduced in [79]. This framework allowed us to discover a new increasing attribute (number of parent nodes) that outperforms the hierarchical watersheds based on the increasing attributes presented in Section 2.7.2, including area, volume and dynamics. In section 6.3, we introduce number of parent nodes. In section 6.4, we present our baseline evaluation method based on precision and recall (for boundaries). In section 6.5, we introduce the region-based assessment measures proposed in [79], which complement the information provided by precision-recall curves. Then, we present experimental results with some morphological hierarchies in section 6.6.

6.2 Cut of a hierarchy

Let $\mathcal{H}$ be a hierarchy of partitions on a set V. Let $\mathbf{P}$ be a partition of V. We say that $\mathbf{P}$ *is a cut of* $\mathcal{H}$ if, for every region R of $\mathbf{P}$, there is a partition $\mathbf{P}'$ of $\mathcal{H}$ such that R is a region of $\mathbf{P}'$. If every region of $\mathbf{P}$ belongs to the same partition of $\mathcal{H}$, *i.e.*, if $\mathbf{P}$ is a partition of $\mathcal{H}$, then we say that $\mathbf{P}$ *is a horizontal cut of* $\mathcal{H}$. The set of regions of $\mathcal{H}$ is

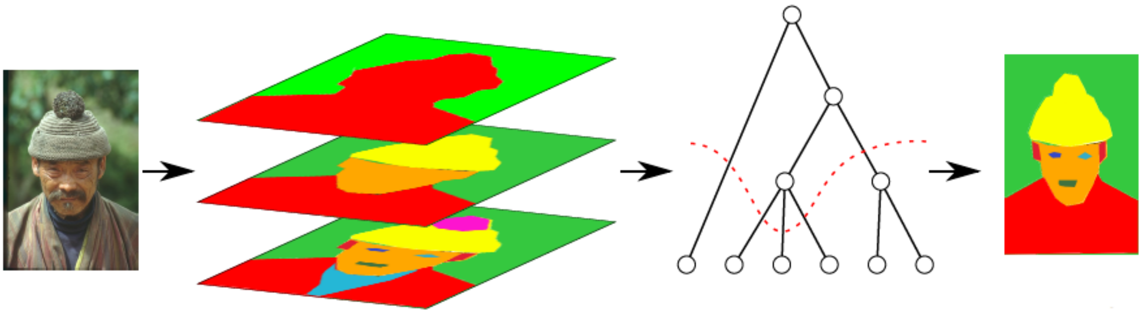
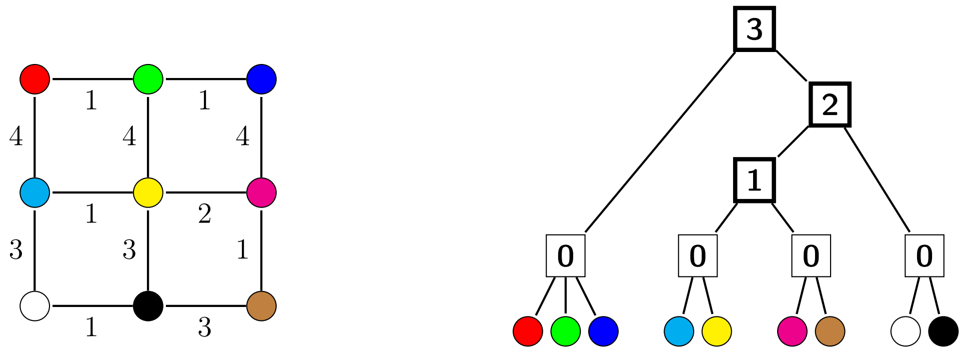


Figure 6.1: A hierarchy of partitions of an image (source BSDS 500 [63]) is a sequence of coarse to fine partitions. The hierarchy can be represented as a tree of regions. A cut is a partition made of regions of the hierarchy possibly taken at different levels.



(a) (G, w) (b) MT

Figure 6.2: (a): a weighted graph (G, w). (b) the min-tree MT of (G, w) (square nodes). The parent nodes of MT are indicated in bold. The number of parent nodes included in each region of MT is indicated inside each node of MT.

the union of all partitions of $\mathcal{H}$. We denote the set of regions of $\mathcal{H}$ by $\mathbf{R}_{\mathcal{H}}$. The set of all cuts of $\mathcal{H}$ is denoted by $\Pi(\mathcal{H})$.

The notion of cuts is illustrated in Figure 6.1.

6.3 Number of parent nodes

Let (G, w) be a graph and let MT be the min-tree of (G, w). As established in Section 2.7.2, increasing attributes and extinction values are computed on the min-tree of (G, w). Let R be a region of MT. The *number of parent nodes of* R is the number of regions of MT that are included in R and that are the parent of at least one region of MT: $|\{R' \subseteq R \mid \exists R'' \in \mathcal{R}_{MT}, parent(R'') = R'\}|$.

In Figure 6.2, we show a weighted graph (G, w), the min-tree MT of (G, w), and the number of parent nodes of each node of MT.

The number of parent nodes of a node measures the number of times the minima of the gradient are modified, either by the addition of new pixels (growth of the associated catchment basin), or by the merging with another minima (fusion of catchment basins). Intuitively, it measures the amount of change in a given region where minima and flat components have been contracted as single pixels. As a topological feature, it is invariant to monotone contrast transformations and to geometric transformations (up to discretization effects). It is also increasing (the attribute value of a node is larger than the one of its children) which allows defining an extinction value associated to each minima of the function [98], thus leading to a hierarchical watershed.

6.4 Baseline: precision-recall for boundaries

Precision and recall are standard evaluation measures of classification problems. Let us consider a binary classification problem in which data are classified as belonging or not to a set or category A. Precision indicates the percentage of the data correctly classified as belonging to A (true positives) among all data classified as belonging to A (true positives + false positives). In turn, recall is the ratio between the data correctly classified as belonging to A (true positives) and the data that belongs to A (true positives + false negatives). The precision P and the recall R measures can be aggregated into the (balanced) F-measure:

$$F = 2 \times \frac{P \times R}{P + R} \tag{6.1}$$

The F-measure values range from 0 to 1, where 1 indicates an ideal classification.

Image segmentation can be formulated as a classification problem. For instance, in [84], the authors propose the following formulation. Let N be the set of pairs of neighboring pixels of an image I and let A be a set composed of elements of N such that, for any element (x, y) in A, we have x and y belonging to the same region of a ground-truth segmentation of I. The precision and recall of a segmentation of I can be obtained by considering the percentage of pair of pixels correctly classified as belonging to the set A. In this context, the precision and recall measures result in the F-measure for regions (FR).

In [63], the authors propose precision and recall measures on segmentation boundaries (FB). They define an *edgel* as a one-pixel fragment of segmentation boundary represented by two parameters: a position in the image plane and an orientation. The precision and recall measures are determined by the matching of edgels in two segmentations. They model this problem through a bipartite graph linking the edgels of two segmentations by weighted edges, whose weights are determine by the euclidean distance and by the difference in orientation of edgels. As the edgel sets of two segemtnations are not likely to have the same cardinality, they insert outlier edges with large weights in the bipartite graph. The edges matched to outlier edgels are considered unmatched. The error is given by the number of unmatched edgels, *i.e.*, the edgels matched to outlier edgels. Their method is robust with respect to segmentations produced by different images: the precision and recall errors of matching edgels from segmentations of distinct images is consistently higher than the errors of segmentations of a same image. It is also sensitive to over and undersegmentation: neither precision nor recall are tolerant to refinement.

The work of [85] on the evaluation of segmentation assessment measures has shown that FB is highly discriminant between ground truths of different images on the BSDS 500 image dataset [63]. On the contrary, FR has shown a low discriminant power. Hence, we consider FB as our baseline segmentation assessment measure.

6.5 Proposed evaluation methodology

In this section, we present the evaluation framework for hierarchies of partitions introduced in [79]. This framework is composed of several supervised assessment measures, each enabling to quantify a different aspect of the hierarchy. The assessment methodology introduced in [79] comprises three parts:

1. An evolution of the upper-bound on region measures [83] enabling to quantify the maximal achievable score of a hierarchy for the general segmentation problem;

2. A new evaluation measures that aims to quantify the easiness of finding a set of regions of a hierarchy representing a semantic object in the scene;

3. The F-measure and precision-recall curves on boundaries FB [64] (see Section 6.4) as a standard evaluation measure. This measure is complementary to the other region oriented measures and also provides a reference measure for the comparison with the literature.

6.5.1 Upper-bound on BCE measure

The horizontal cuts considered in the framework of [85] represent a subset of all possible partitions that can be constructed from a hierarchy. In order to better evaluate the potential of hierarchies, the authors of [82, 83] proposed to look for the optimal cut, generally not horizontal, in a hierarchy according to a given evaluation measure.

We propose an evolution of the evaluation on regions proposed by Pont-Tuset *et al.* [83] that consists in two improvements: 1) the use of a dissimilarity measure that enables to penalize both under- and over-segmentation, and 2) the definition of a new type of curve, the fragmentation upper-bound curve that enable to measure the potential of the hierarchy and the potential gain of non-horizontal cuts compared to horizontal cuts. In [83], the authors focused on the directional Hamming distance [50] which is transparent to over-segmentation, i.e., it does not penalize the subdivision of a region of the ground-truth into multiple regions in the proposal segmentation. We propose to use the FB (see Section 6.4) and the Bidirectional Consistency Error BCE [64] measure. The BCE measure is symmetric and it is not transparent to over- or undersegmentation. The evaluation of segmentation measures provided by [85] evaluates BCE as a highly discriminant measure on the image segmentation dataset BSDS500 [63].

Given an image I, one ground-truth segmentation $\mathbf{T}_I$, and a proposal segmentation $\mathbf{S}_I$, the *BCE measure of* $\mathbf{S}_I$ *and* $\mathbf{T}_I$ is defined by [64]:

$$BCE(\mathbf{S}_I, \mathbf{T}_I) = \frac{1}{N} \sum_{R \in \mathbf{S}_I, R' \in \mathbf{T}_I} |R \cap R'| min \left(\frac{|R \cap R'|}{|R|} \frac{|R \cap R'|}{|R'|} \right) \tag{6.2}$$

Given a similarity measure s, an image I, one ground-truth segmentation $\mathbf{T}_I$, and a proposal segmentation $\mathbf{S}_I$, we denote by $s(\mathbf{S}_I, \mathbf{T}_I)$ the similarity between $\mathbf{S}_I$ and $\mathbf{T}_I$ for s. Given a hierarchy of partitions $\mathcal{H}_I$ on the image I, one ground-truth segmentation $\mathbf{T}_I$ and a number k of regions, the *Upper-Bound score for s* (UB_s) for $\mathcal{H}_I$ is the highest score according to s for all the cuts of $\mathcal{H}_I$ composed of k regions:

$$UB_s(\mathcal{H}_I, \boldsymbol{T}_I, k) = \max_{\substack{\boldsymbol{S} \in \boldsymbol{\Pi}(\mathcal{H}_I) \\ |\boldsymbol{S}| = k}} s(\boldsymbol{S}, \boldsymbol{T}_I). \tag{6.3}$$

In order to better understand the content of the hierarchies and to account for the variations inside the evaluation datasets, we propose the Fragmentation–Optimal Cut score curve (FOC) where the mean-average Upper-Bound BCE score (the mean image score over the database, with the image score defined as the average score over the set of ground-truths for the image) is plotted against the fragmentation level of the segmentation defined as $k/|\mathbf{T}_I|$, the ratio between the number of regions in the segmentation and the number of regions in the ground-truth (see Figure 6.3). The gain achieved by taking a non horizontal cut in the hierarchy is evaluated with a second curve: the Fragmentation-Horizontal Cut score curve (FHC) obtained by taking the successive partitions of the

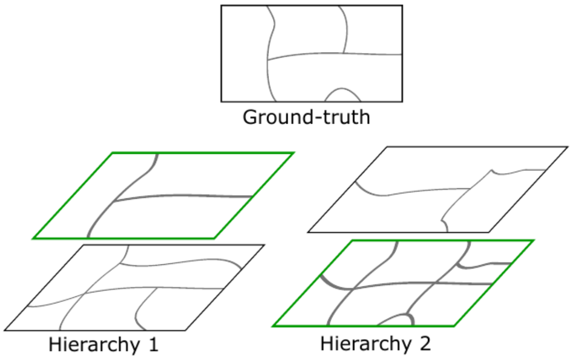


Figure 6.3: Illustration of under- and over-segmentation for hierarchies. Hierarchies 1 and 2 are both composed of 2 levels. Compared to the ground-truth, the first hierarchy manages to recover long contours in its coarse level but then fails to recover the other contours at a finer level: the optimal horizontal cut is the coarsest one and the hierarchy is said to under-segment the image. With the second hierarchy the inverse situation happens, the coarsest partition recovers all the contours of the ground-truth but also contains extra-contours. However, the finest partition loses the true contours and preserves extra contours: the hierarchy is said to over-segment the image.

hierarchy (similarly to precision-recall curves). A large difference between the FOC and FHC curves suggests that the optimization algorithm has selected regions from various levels of the hierarchy to find the optimal cut: the regions of the ground-truth segmentations are thus spread at different levels in the hierarchy.

The FOC curve starts at the value corresponding to the single region partitions (independent of the evaluated hierarchy). Then, it generally quickly increases at low fragmentation levels as the optimization first selects the largest regions that summarize the ground-truth. Then, the optimal cut starts to include smaller regions that provides only little score gain: this corresponds to the nearly flat part of the curve. At a high level of fragmentation (not visible in the figures), the algorithm cannot add new regions without lowering the score and the curve starts to decrease.

In the ideal case, the maximum of the FOC and FHC curves is achieved for a fragmentation of 1. If the maximum happens at fragmentation level lower than 1, this means that the hierarchy tends to capture the main feature of the ground-truth with a low number of regions but then fails to correctly refine those regions (see Hierarchy 1 in Figure 6.3): in this case we say that the hierarchy has a tendency for under-segmentation. If the maximum happens at a fragmentation level higher than 1, this means that the hierarchy is able to provide a set of superpixels for the ground-truth but fails to merge them in a correct order (see Hierarchy 2 in Figure 6.3): in this case we say that the hierarchy has a tendency for over-segmentation.

As an overall performance summary respectively on the FOC and FHC curves, we compute the normalized area under the curve, denoted respectively by AUC-FOC and AUC-FHC. The area under the curve provides an evaluation over a large range of fragmentation levels and thus accounts for the hierarchical nature of the object of study. In order to obtain a measure that is symmetric between under- and over- fragmentation, the area under the curve is calculated on the interval $]0, 2]$. Finally, the area under the curve is normalized with a factor 1/2 to obtain a score between 0 (worst) and 1 (best).

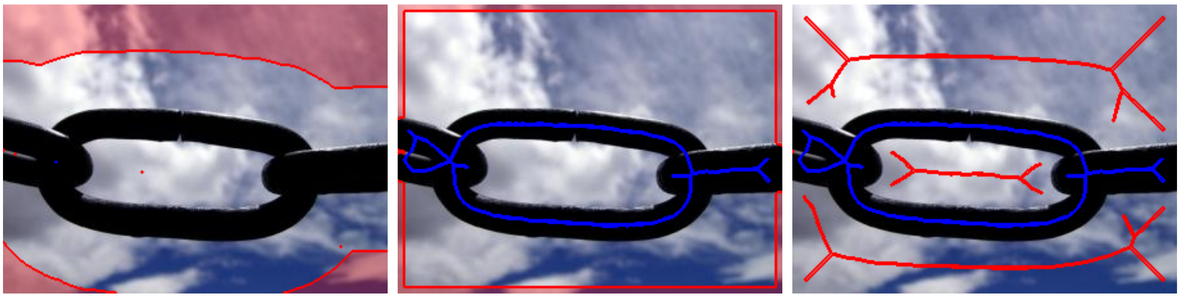
6.5.2 Object detection measure

The last measure, introduced in [80] and studied in [79], is based on supervised object detection with markers. It quantifies how well a specific object of a scene can be retrieved with different levels of information given on its position.

We use the procedure described in [87] that constructs a two- class segmentation from a hierarchy of partitions and two non-empty markers: one for the background and one for the object of interest. Its principle is to identify the object as the union of the regions of the hierarchy that intersect the object marker but does not touch the background marker. Formally, given an image I, a hierarchy $\mathcal{H}_I$, an object marker M_o, and a background marker M_b, the extracted object is defined by:

$$O(\mathcal{H}_I, M_o, M_b) = \bigcup\{R \in \mathbf{R}_{\mathcal{H}_I} \mid R \cap M_o \neq \emptyset, R \cap M_b = \emptyset\}. \quad (6.4)$$

This result can be computed efficiently with the following algorithm. In the first step of the algorithm, the hierarchy is browsed from the leaves to the root. If the current node is labeled *Background* then its parent node intersects the background marker and is labeled *Background*. If the current node is labeled *Object* and its parent is not currently labeled then it can be labeled *Object*. In the second step, the tree is browsed from the root to the leaves and any non labeled node takes the label of its parent. Finally, the labels of the leaves (the image pixels) give the segmentation result. In order to perform an objective assessment of the different hierarchies we propose several automatic strategies to generate object and background markers from the ground truths. The main idea is not to reproduce the interactive segmentation process experienced by a real user but rather to obtain markers representing different difficulty levels or that resembles to human generated markers. The generated markers are the following (see Figure 6.4): 1) Erosion (Er): erosion by a ball of radius 45 pixels. If a connected component is completely deleted by the erosion then a single point located in the ultimate erosion of this connected



(a) Er-Er (b) Sk-Fr (c) Sk-Sk

Figure 6.4: Different combinations of markers. The combination of markers is indicated in the caption of each sub-figure in the form Background Marker-Object Marker. In each figure the background and object markers are respectively depicted in red and blue.

component is added to the marker, 2) Skeleton (Sk): morphological skeleton given by [19], and 3) Frame (Fr): frame of the image minus the object ground truth if the object touches the frame (background only). Using the frame as the background marker is nearly equivalent to having no background marker in the sense that it does not depend of the ground truth or of the image. In the following, the combination of the background marker MB and the object marker MF is denoted MB-MF (for example, Fr-Sk stands for the combination of a Frame marker for the background and a skeleton marker for the object). Among all the possible combinations of markers, we chose to concentrate on the following ones: 1) Sk-Sk resembles to human generated markers, 2) Er-Er leaves a large space between markers and represent a difficult case. Nevertheless, the combination is symmetric in the sense that the correct segmentation is roughly at equal distance from the object and from the background marker, and 3) Fr-Sk where the object marker resembles to a human generated marker and the background marker conveys nearly no information: this case is thus strongly asymmetric. The performance of each segmentation result is evaluated with the F-Measure. The median score for the 3 marker combinations is called Object Detection Median and is denoted ODM.

6.6 Experiments

This section presents the results of the experiments and some discussions.

Precision-recall curves for boundaries and upper-bound on BCE measure are evaluated on the BSDS500 dataset [63] (200 test images). The object detection measure is evaluated on the Grabcut [16] and Weizmann 1 object [1] datasets (respectively 50 and 100 test images). We study the importance of the gradient measure for all methods (Section 6.6.1) and the necessity to perform a filtering of some hierarchies (Section 6.6.2).

The overall results are discussed and compared to high quality approach (Section 6.6.3).

In the remainder of this section, the hierarchical watersheds based on area, dynamics, volume and parent nodes are respectively denoted by WS-Area, WS-Dynamics, WS-Volume and WS-Parents. The quasi-flat zones hierarchies, revised in Section 2.3, are denoted by QFZ.

6.6.1 Influence of the gradient

A classical way to weight the edges of a graph in image analysis in general and for morphological segmentation in particular is to use a gradient measure. The aim of this section is to evaluate the influence of the gradient measure on the quality of the hierarchies.

The most simple gradient measures use only colorimetric information from the two pixels of an edge: in this category, we consider an Euclidean distance in the RGB color space and an Euclidean distance in the Lab color space, the latter being more compliant with human color perception. However, recent advances on contour detection have led to non local supervised gradient estimators achieving better performance on contour detection benchmarks: in this category, we consider the globalized probability of boundary (gPb) from [6] and the structured edge detector (SED) from [29].

Figure 6.5 shows the result of WS-Dynamics (top row) and WS-Area (bottom row) with the four considered gradients–RGB, Lab, gPb, and SED. The results of QFZ (respectively WS-Volume and WS-Parents), not shown here, are similar to the results of WS-Dynamics (respectively WS-Area). A first observation on WS-Dynamics with RGB and Lab gradients is that its PR-Curves on boundaries seem truncated and its FOS-Curves on regions are flat. In the first case, the truncation appears at the level where the partition of the hierarchy contains more than 3000 regions: the evaluation procedure is stopped at this point as it becomes two demanding on computational power. In the second case, the flat curve is the result of the hierarchy not being able to provide any meaningful partition with at most twice the number of regions in the ground-truth. Those two observations can be a consequence of WS-Dynamics (and similarly QFZ) having its upper levels made only of small salient regions; a solution to this problem is presented in the next section.

While the Lab gradient provides slightly better performance compared to RGB gradient in most cases, we observe a large gain by switching from a local RGB or Lab gradient to a supervised non-local gradient like gPb or SED. The SED gradient improves the results for every measure except the FOC curve with WS-Dynamics compared to gPb gradient. The FOC and FHC curves show that WS-Dynamics requires much more regions to reach its maximal scores with SED gradients which implies that the hierarchy

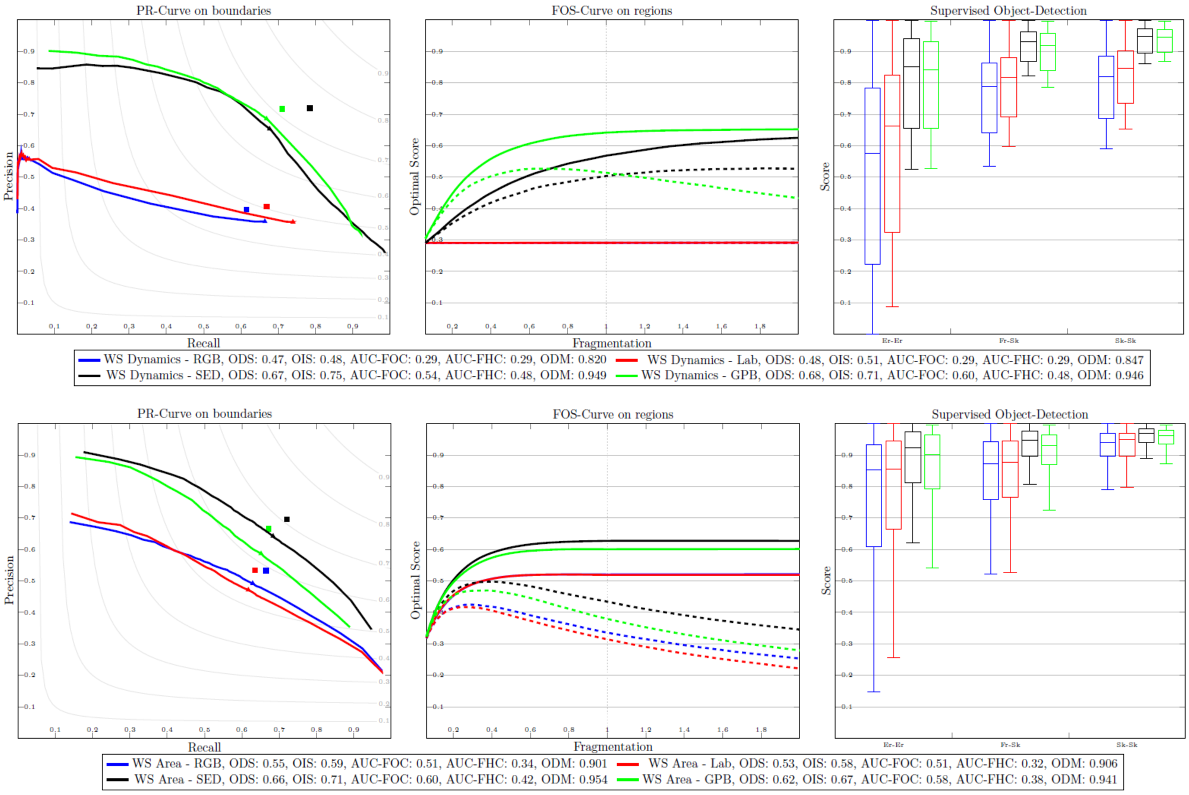


Figure 6.5: Influence of the gradient on WS-Dynamics (top row) and WS-Area hierarchies (bottom row). **Mean Precision-recall (PR) curves for boundaries** on BSDS500: each curve represents the variation of precision and recall for the different partitions of the hierarchy. OIS and ODS scores are given in the legend and are respectively represented in the plot by a square and a triangle. **Fragmentation–Optimal Cut score curves (FOC)** for regions on BSDS 500: each plain curve represent the upper-bound score achievable for a given fragmentation value. The corresponding dashed curves represent the score obtained by horizontal cuts. Area under curve for the plain curve (AUC-FOC) and the dashed curve (AUC-FHC) are given in the legend. **Supervised object detection on Grabuct and Weizmann datasets:** for each method and each combination of markers, we see: 1) the median F-measure (central bar), 2) the first and third quartile (extremities of the box), and 3) the lowest datum still within 1.5 inter quartile range (difference between the third and first quartile) of the lower quartile, and the highest datum still within 1.5 inter quartile range of the upper quartile range (lower and upper extremities). The median score over all markers combinations is given in the legend.

tends to have small irrelevant regions on its top layers. This suggests that despite the regularization effect based on dynamics, which tends to send lowly contrasted regions to the lower levels of the hierarchy, WS-Dynamics remains sensitive to small regions of high contrast that appear more often in SED gradients than in gPb gradients.

In conclusion, we recommend the use of SED gradient to build hierarchical watersheds on natural images and the following experiments will be conducted with this gradient. Moreover, SED is about 3 orders of magnitude faster than gPb [6] enabling to reach real time performance without any particular material.

6.6.2 Small regions removal

As observed in the previous section, QFZ and WS-Dynamics are sensitive to small regions even with a smooth gradient as SED. In this section we evaluate the impact of an area post-filtering on those hierarchies.

The area filter described in [47] removes contours iteratively in the hierarchy: starting from the leaves and moving toward the root, the children of a node are merged if at least one of them contains less than k pixels. In the following, we express the strength of the filter as the ratio $r_k = k \setminus N$, with N the number of pixels in the considered image.

Figure 6.6 shows the result of the filtering on QFZ (the results on WS-Dynamics are similar) with four different values of r_k: 0 (no filter), 0.4‰ (roughly 50 pixels in a BSDS500 image [63]), 0.8‰, and 1.6‰. We observe that all measures increase with r_k, from $r_k = 0$ to $r_k = 0 : 8‰$. The introduction of the filtering immediately produces a large performance boost. For $r_k = 1.6‰$ compared to $r_k = 0 : 8‰$, the situation is mixed with an improvement on FOC measures, stagnation on objection detection measures, but a degradation of OIS and ODS scores: this reflects a tradeoff between the number of regions necessary to describe the scene and the precision of boundaries.

The effect of the filtering on under- and oversegmentation is presented in Figure 6.7. For each image and each ground-truth of the dataset, we plot the number of regions present in the optimal segmentation found against the number of regions in the ground-truth (we define the optimal segmentation as the segmentation that achieves 99% of the optimal score with the fewest number of regions in the FOC curve). We see that for WS-Dynamics (results are similar for QFZ), larger values of r_k tends to push the optimal segmentation from over-segmentation (position above the diagonal where the optimal segmentation contains more regions than the ground-truth) to under-segmentation (position below the diagonal where the optimal segmentation contains less regions than the ground-truth). For $r_k = 0 : 8‰$, the optimal solutions have a mostly symmetrical distribution around the diagonal, suggesting no bias toward under or over-segmentation.

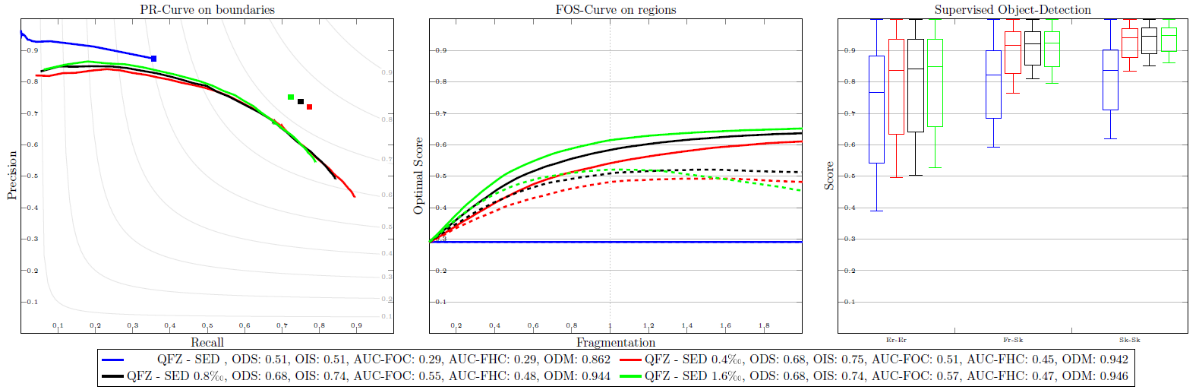


Figure 6.6: Influence of the area filter on QFZ. (See Figure 6.5 for explanation).

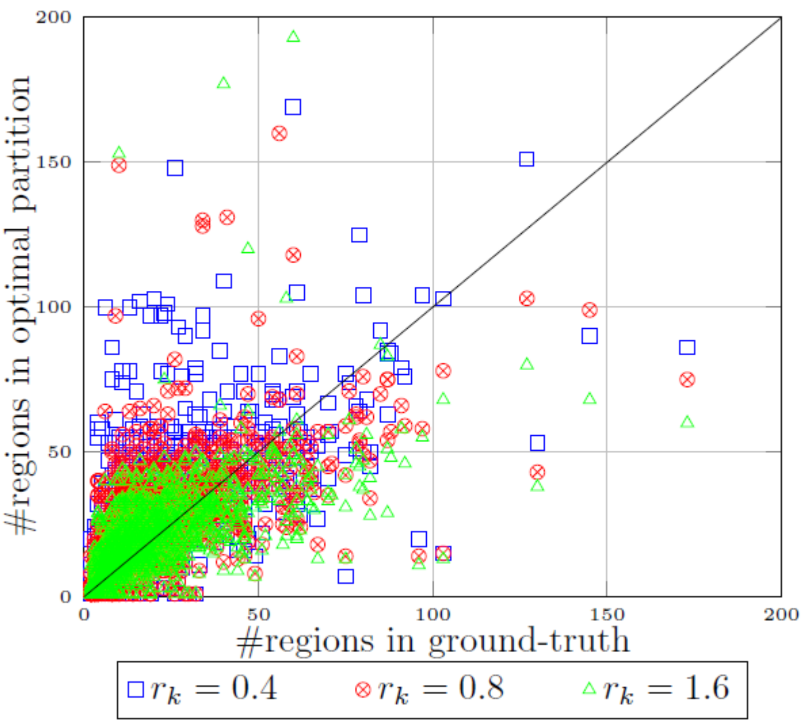


Figure 6.7: Influence of area filtering on WS-Dynamics.

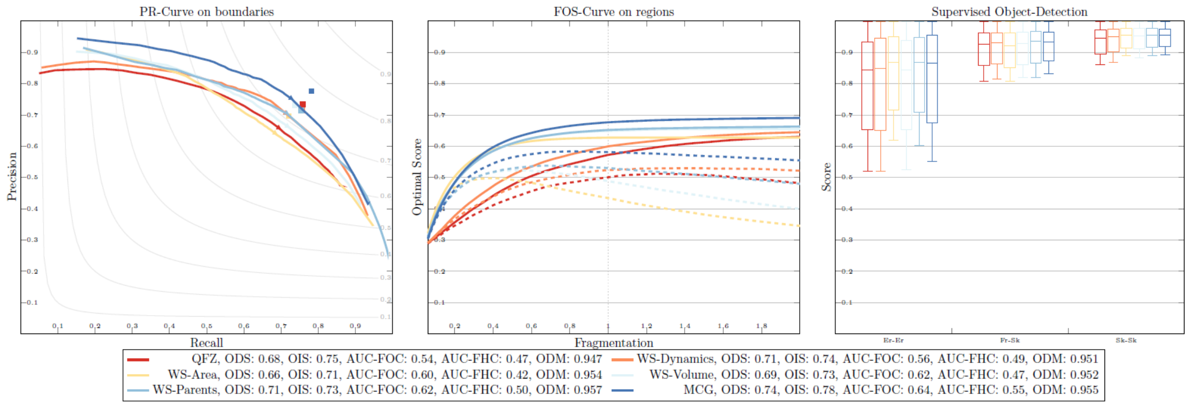


Figure 6.8: Best achieved results for each hierarchy and a high quality hierarchical segmentation methods. (See Figure 6.5 for explanation).

In conclusion, for QFZ and WS-Dynamics we recommend to perform a post-filtering of the hierarchy by removing regions smaller than 0.8‰ of the image size. One can notice that the object detection measure is less sensitive to the area filtering than other evaluation measures. This suggests that, for some applications, the processing of the hierarchy is naturally robust to small nodes and this filtering may not be necessary.

6.6.3 Discussions

This section compares and discusses the best results obtained for each hierarchy (see Figure 6.8).

As a reference we also include the Multiscale Combinatorial Grouping (MCG) hierarchies from [81] in our assessments. MCG also uses SED as the main cue for contour detection, but then merges several hierarchies (referred as OWT-UCM in the literature [6]) computed at different scales.

We can observe that QFZ is globally inferior to all the other methods. WS-Dynamics shows good performances in precise contour placement (high ODS and OIS scores) but has a clear tendency for over-segmentation (maximum of FOC and FHC curve occur at large values of fragmentation). We can also notice that WS-Dynamics (and QFZ) performances for object detection in the Sk-Sk case is significantly lower than other methods; this suggests that the hierarchy fails to correctly order regions near the boundaries which is coherent with its tendency to over-segment (more regions are needed to obtain the true contours).

On the contrary, WS-Area and WS-Volume show weaker performance at contour location (average OIS and ODS scores) and have a tendency for under-segmentation (maximum of FOC and FHC curve occur at low values of fragmentation). WS-Area shows a clear advantage over other methods on object detection with Er-Er markers which can be explained by the symmetric nature of the markers in this case: the true contour is located roughly at equal distance from both markers and WS-Area is particularly good at producing a regular (in size) tiling of the contour image.

WS-Parents offers the best performances on every measure expect OIS compared to other hierarchical watersheds. As WS-Area and WS-Volume, it shows a small tendency for under-segmentation. The results of MCG remain higher than WS-Parents except for the object detection assessment: this suggests that MCG, whose various components have been either trained or optimized on BSDS500 dataset, may over-fit this particular dataset and not be the best method for other applications than general segmentation.

6.7 Conclusion

We presented a novel evaluation framework for the evaluation of hierarchies of partitions that enables to capture the quality of different aspects of the hierarchies: regions, contours, horizontal cuts, optimal cuts, nodes grouping, under or over-segmentation. Compared to the classical approach for hierarchy evaluation that concentrates only on the horizontal cuts and the image segmentation problem, we believe that the proposed framework offers a richer assessment that better accounts for the hierarchical nature of the representation and is not limited to a single use case.

This framework was used to assess various hierarchies of morphological segmentations. In particular, we studied the importance of the gradient measure for all methods and the necessity to perform a filtering of some hierarchies. The framework also allowed us to identify a hierarchical watershed based on a novel extinction value, the number of parent nodes, that outperforms the other hierarchies of morphological segmentations. We have shown that, used in conjunction with a state-of-the art contour detector, most hierarchical watersheds are competitive or even sometimes better than the complex state of the art method for hierarchy construction. Moreover, hierarchical watersheds are well defined structure satisfying clear global optimality properties and can be computed efficiently on large data: they are thus valuable candidates for various computer vision tasks.

Chapter 7

Combination of hierarchies

In this chapter, we present a general framework to combine (the saliency maps of) hierarchies of partitions. We first assess this framework on combinations of hierarchical watersheds and of hierarchies based on non-increasing attributes. Then, we study properties of combinations of hierarchical watersheds based on the characterization of hierarchical watersheds and on the notion of watersheding operator presented in chapters 3 and 4, respectively. This chapter comprises the results of the following articles:

- D. S. Maia, A. de Albuquerque Araujo, J. Cousty, L. Najman, B. Perret, and H. Talbot. Evaluation of combinations of watershed hierarchies. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 133–145. Springer, 2017.
- D. S. Maia, J. Cousty, L. Najman, and B. Perret. Properties of combinations of hierarchical watersheds. Under review. 2019.
- D. S. Maia, J. Cousty, L. Najman, and B. Perret. Characterization of graph based hierarchical watersheds: theory and algorithm. Under review. 2019.

7.1 Introduction

In the construction of a hierarchy of segmentations, two aspects should be considered: (1) the regions that compose the finest partition of the hierarchy; and (2) the criterion under which those regions are merged along the hierarchy. In this study, we focus on the second aspect of the construction of hierarchies. More precisely, we explore the method to combine hierarchies introduced in [28, 25]. In [28, 25], the authors combine the saliency maps of hierarchies that are connected for a same graph. The input saliency maps are combined into a new map. Then, the combined hierarchy is the QFZ hierarchy of the resulting map. Other approaches on combinations of hierarchies of segmentations have already been tackled in [52, 33, 59, 76].

In [52], the authors propose to fuse hierarchies of partitions with other functions that are not necessarily saliency maps. They formalize a segmentation as a set of Jordan curves and then explore the lattice induce by this set of Jordan curves, along with the opening and closing operators defined on this lattice. Those operators induce the definition of supremum and infimum of two functions and of two hierarchies. In their experiments, they combine the saliency maps of hierarchies of partitions with functions associated to a ground-truth segmentation, both defined on a 2D plane. More precisely, they compute the inverse of the distance map of the ground-truth contours, assigning the largest values of the points in the contours. The resulting combination is the closest to both the ground truth and the initial saliency map of the hierarchy.

In [33], the authors perform a sequential combination of stochastic watershed hierarchies. Given a weighted graph (G, w), they compute the saliency map w' of the watershed hierarchy based on a given attribute A. Then, the graph (G, w') is used to compute another watershed hierarchy based on an attribute A'. We can see that those combinations can be performed indefinitely for any number of attributes. In [34], the authors conclude that sequential combinations are not commutative and that the sequential combinations with a single attribute are convergent.

In [59], the authors use a Convolutional Neural Network (CNN) to build hierarchies of segmentations. They extract oriented image boundaries at different scales from different levels of the CNN. At each level, they compute an ultrametric contour map (saliency map), which are further combined following the approach of [81]: the coarse boundaries obtained from higher level scales are approximated into the finer and better placed boundaries of lower levels.

In [76], the authors compute hierarchical watersheds based on combined attributes. They propose a contact based attribute function that minimizes the contact surface of the final 3D segmented objects. They show the improvement of combining this attribute to the dynamics and volume attributes in the segmentation of grains of 3D tomography images.

In general, no single hierarchy of segmentations completely fits the applicative and cognitive expectation for all regions of an image. This is illustrated in Figure 7.1, where we present a combination of two hierarchical watersheds obtained through the method described in [25]. Those hierarchies are driven by regional attribute values based on area [70] and on dynamics [75]. In Figure 7.1, we also present segmentations with 75

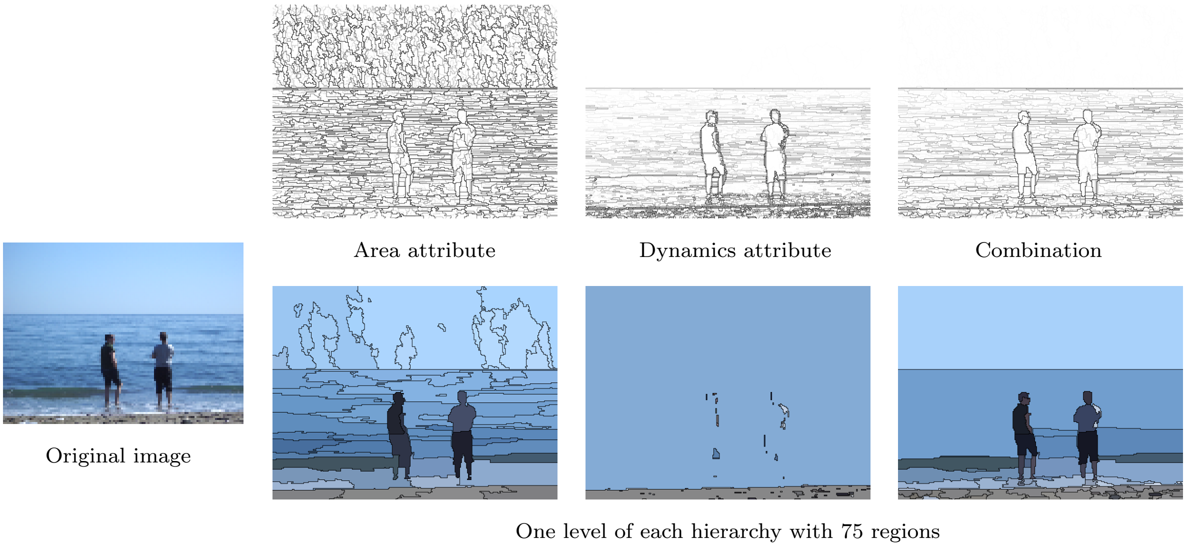


Figure 7.1: Hierarchical watersheds based on area and dynamics and their combination by average.

regions extracted from each hierarchy. Hierarchies are represented thanks to their saliency map. In this representation of saliency maps, the darkest contours are the ones that persist at the highest levels of the hierarchies. From the saliency map of the hierarchical watershed based on area shown in Figure 7.1, we can see that the sky and sea regions are oversegmented at high levels of this hierarchy. On the other hand, spurious regions are preserved at high levels of the hierarchy based on dynamics and the two people, which are significant regions from a cognitive point of view, are merged with the background. We can see that, in the combination of those two hierarchies, the two people are preserved at high levels of this hierarchy and the sky and sea regions are less oversegmented. Based on this visual inspection and on the illustrations of combinations of hierarchical watersheds provided in [25], we expect combinations of hierarchies to perform better than the initial hierarchies.

In this chapter, we explore the potential of combinations of hierarchies. It is organized as follows:

- Section 7.2, reviews the combination framework of hierarchies introduced in [28, 25]: given two hierarchies, their saliency maps are combined through an edge-wise function and, then, the resulting hierarchy is the QFZ hierarchy of the combined saliency maps;

- Section 7.3 proposes a method to obtain normalized saliency maps. For a combination of hierarchies to make sense, we need their saliency maps to convey similar information at each level, *e.g.* to have the same number of regions at each level. Hence, we treat the cases where the input saliency maps do not have the same number of level-sets, and the cases where the range of the saliecy maps differ;

- Section 7.4 presents a visual inspection of combinations of hierarchical watersheds and of combinations of hierarchies based on non-increasing attributes. We show that some of the combinations successfully bring to the fore the most important regions of an image.

- Section 7.5 presents a quantitative assessment of combinations of hierarchical watersheds using the evaluation framework presented in Chapter 6. We compare all combinations of hierarchies with the initial hierarchies, and with the hierarchical watershed of best performance, namely the hierarchical watersheds based on the number of parent nodes (see Section 6.6.3). Nearly half of the combinations outperformed the initial hierarchies. Moreover, some combinations also outperformed the hierarchical watersheds based on the number of parent nodes;

- Section 7.6 studies properties of combinations of hierarchical watersheds. More precisely, we investigate the following problem: do combinations of hierarchical watersheds result in hierarchical watersheds? We conclude that the combinations of hierarchical watersheds with the combining functions studied here do not result in hierarchical watersheds, in general. We also provide a sufficient condition for a combining function to always output flattened hierarchical watersheds (see Definition 21);

- Section 7.7 presents experiments with our algorithm to recognize (flattened) hierarchical watershed (see Chapter 3) applied to combinations of hierarchies. We conclude that, when hierarchical watersheds are computed by considering arbitrary orderings on the edges of a graph, nearly one third of the combinations do not result in flattened hierarchical watersheds. However, when the algorithm to compute hierarchical watersheds considers a unique ordering on the edges, virtually all combinations are flattened hierarchical watersheds;

- Section 7.8 presents the results of the watersheding operator (see Chapter 4) applied to combinations of hierarchical watersheds. We observe that, in most cases, the watersheding of combinations of hierarchies improves the combinations in terms of quantitative evaluation, with the advantage of outputting hierarchies which are optimal in the sense of MSFs, as discussed in Section 2.6.3;

- Finally, we conclude and discuss future work in Section 7.9.

Important notations: in the remainder of this chapter, the symbol (G, w) denotes a weighted graph whose vertex set is connected. To shorten the notation, the vertex set of G is denoted by V and its edge set is denoted by E. Every hierarchy considered in this chapter is connected for G and therefore, for the sake of simplicity, we use the term *hierarchy* instead of *hierarchy which is connected for G*. Given an altitude ordering $\prec$ for w and a building edge u for $\prec$ (see definitions in Section 2.5.2), we denote by R_u the region of the binary partition hierarchy $\mathcal{B}_\prec$ by $\prec$ whose building edge is u. The set of building edges for $\prec$ is denoted by $E_\prec$. We denote by n the number of minima of w. Every sequence of minima of w considered in this chapter is a sequence of n pairwise distinct minima of w and, therefore, for the sake of simplicity, we use the term *sequence of minima of w* instead of *sequence of n pairwise distinct minima of w*. By abuse of terminology, when no confusion is possible, if M is a minimum of w, we call the set $V(M)$ of vertices of M as a minimum of w.

7.2 General combination framework

Combining partitions and, *a fortiori*, hierarchies is not straightforward. This problem has been tackled in [52, 58, 28, 25] thanks to the use of saliency maps and we follow the same approach. More precisely, in order to combine two hierarchies $\mathcal{H}_1$ and $\mathcal{H}_2$, we proceed in three steps: first the saliency maps of $\mathcal{H}_1$ and $\mathcal{H}_2$ are considered, then the two saliency maps are combined to obtain new weights on the edges of G, and, finally, the combination of hierarchies is the QFZ hierarchy of the new map (see Figure 7.2).

Let $\mathcal{F}$ be the set of all maps from E into $\mathbb{R}$. Any map C from $\mathcal{F}^2$ into $\mathcal{F}$ is called a *combining function*.

Given two hierarchies $\mathcal{H}_1$ and $\mathcal{H}_2$ and a combining function C, the *combination of* $\mathcal{H}_1$ *and* $\mathcal{H}_2$ *by* C is the hierarchy $\mathcal{H}_C(\mathcal{H}_1, \mathcal{H}_2)$ defined by:

$$\mathcal{H}_C(\mathcal{H}_1, \mathcal{H}_2) = \mathcal{QFZ}(C(\Phi(\mathcal{H}_1), \Phi(\mathcal{H}_2))). \tag{7.1}$$

We consider three classical functions in the instantiation of the combining function (supremum, infimum and linear combination), and a new function called *concatenation*. Given two maps f and g in $\mathcal{F}$, the supremum, infimum and linear combination of f and g, respectively denoted by $\curlyvee(f, g)$, $\curlywedge(f, g)$ and $\boxplus_\alpha(f, g)$, are defined for each edge u in E as:

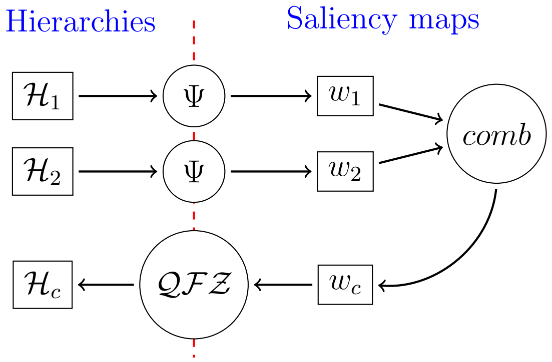


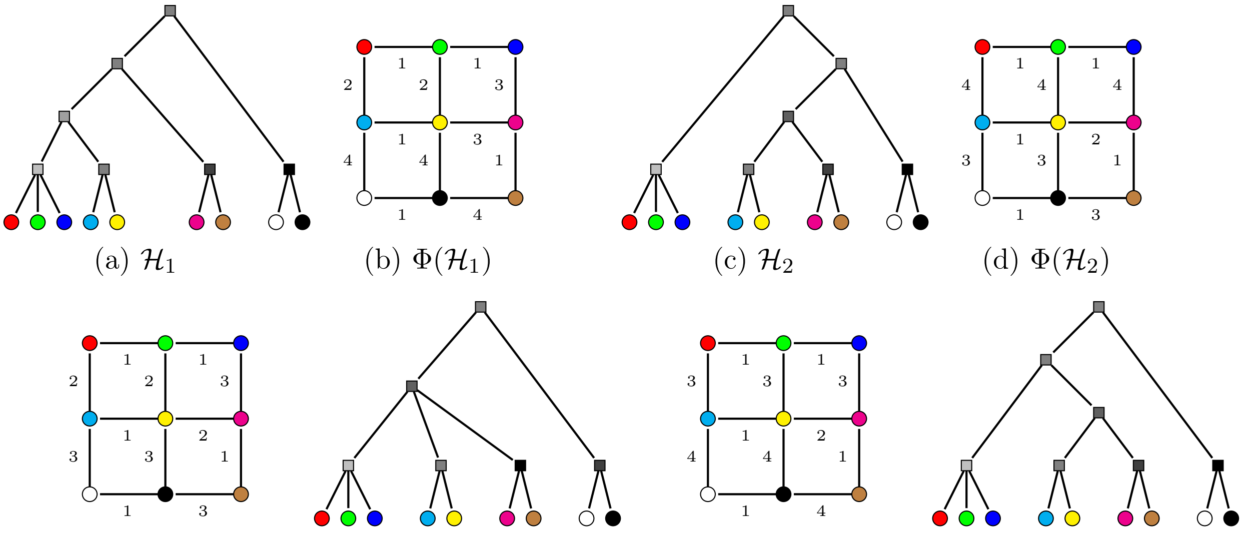
Figure 7.2: Scheme of the method to combine hierarchical watersheds investigated here. First, given two hierarchical watersheds $\mathcal{H}_1$ and $\mathcal{H}_2$, the saliency maps w_1 and w_2 of respectively $\mathcal{H}_1$ and $\mathcal{H}_2$ are computed. Then, the saliency maps w_1 and w_2 are combined, resulting in the map w_c. Finally, the resulting hierarchy is the quasi-flat zones hierarchy of w_c.

$$\begin{aligned} \curlyvee(f,g)(u) &= max(f(u),g(u)) \\ \curlywedge(f,g)(u) &= min(f(u),g(u)) \\ \boxplus_\alpha(f,g)(u) &= \alpha f(u) + (1-\alpha)g(u) \end{aligned} \tag{7.2}$$

where the real-valued parameter α is in the range $[0,1]$.

One example of a combination of hierarchies by infimum is shown in Figure 7.3. Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be the hierarchies of Figure 7.3(a) and (c), respectively. To combine $\mathcal{H}_1$ and $\mathcal{H}_2$ by infimum, we consider their saliency maps $\Phi(\mathcal{H}_1)$ and $\Phi(\mathcal{H}_2)$ illustrated in Figure 7.3(b) and (d), respectively. The edge-wise combination $\curlywedge(\Phi(\mathcal{H}_1), \Phi(\mathcal{H}_2))$ of $\Phi(\mathcal{H}_1)$ and $\Phi(\mathcal{H}_2)$ by infimum is given in Figure 7.3(e). The combination of $\mathcal{H}_1$ and $\mathcal{H}_2$ by infimum, shown in Figure 7.3(f), is the QFZ hierarchy of the map $\curlywedge(\Phi(\mathcal{H}_1), \Phi(\mathcal{H}_2))$.

The purpose of the concatenation is to combine higher levels of a hierarchy with lower levels of another hierarchy. This type of combination is useful when a hierarchy $\mathcal{H}_1$ succeeds at describing the small details of an image at lower levels, but fails at filtering the small regions to capture the main large objects at higher levels of the hierarchy. Therefore, it can be interesting to concatenate $\mathcal{H}_1$ with another hierarchy $\mathcal{H}_2$ whose high levels describe well the important regions in the image. This general idea is represented in Figure 7.4. Given two hierarchies $\mathcal{H}_1$ and $\mathcal{H}_2$, we aim to obtain a new hierarchy $\mathcal{H}_3$ whose high (resp. low) levels correspond approximately to the high (resp. low) levels of $\mathcal{H}_1$ (resp. $\mathcal{H}_2$). In order to define the concatenation of hierarchies, we first define the double threshold function. Given any map f in $\mathcal{F}$ and given two parameters α and β in



(a) $\mathcal{H}_1$ (b) $\Phi(\mathcal{H}_1)$ (c) $\mathcal{H}_2$ (d) $\Phi(\mathcal{H}_2)$

(e) $\curlywedge(\Phi(\mathcal{H}_1), \Phi(\mathcal{H}_2))$ (f) $\mathcal{H}_\curlywedge$ (g) $\uplus_{(2)}(\Phi(\mathcal{H}_2), \Phi(\mathcal{H}_1))$ (h) $\mathcal{H}_{\uplus_3}$

Figure 7.3: (a) a hierarchy $\mathcal{H}_1$. (b) the saliency map $\Phi(\mathcal{H}_1)$ of $\mathcal{H}_1$. (c) a hierarchy $\mathcal{H}_2$. (d) the saliency map $\Phi(\mathcal{H}_2)$ of $\mathcal{H}_2$. (e) the combination $\curlywedge(\Phi(\mathcal{H}_1), \Phi(\mathcal{H}_2))$ of the saliency maps $\Phi(\mathcal{H}_1)$ and $\Phi(\mathcal{H}_2)$ by infimum. (f) the combination of $\mathcal{H}_1$ and $\mathcal{H}_2$ by infimum, which corresponds to the QFZ hierarchy of $\curlywedge(\Phi(\mathcal{H}_1), \Phi(\mathcal{H}_2))$. (g) the combination of the saliency maps $\Phi(\mathcal{H}_1)$ and $\Phi(\mathcal{H}_2)$ with concatenation at level $\lambda = 2$. (h) the concatenation of $\mathcal{H}_1$ and $\mathcal{H}_2$ at level 3.

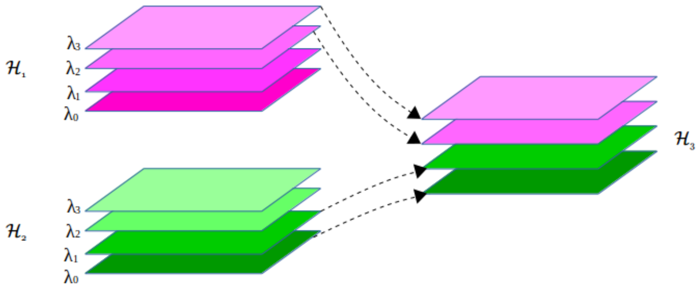


Figure 7.4: Concatenation of low levels of a hierarchy $\mathcal{H}_1$ with high levels of a hierarchy $\mathcal{H}_2$.

$\mathbb{R}$ such that $\alpha < \beta$, we denote by $T(f, \alpha, \beta)$ the *double threshold of f by (α, β)* such that, for any edge u of G:

$$T(f, \alpha, \beta)(u) = \begin{cases} 0 & if\ f(u) < \alpha \\ \beta & if\ f(u) > \beta \\ f(u) & otherwise \end{cases} \tag{7.3}$$

Let f and g be two maps in $\mathcal{F}$. Given a threshold value λ, the *concatenation of f*

and g *at level* λ, for any edge u, is given by:

$$\uplus_\lambda (f, g)(u) = max(T(f, 0, \lambda)(u), T(g, \lambda, \infty)(u)) \quad (7.4)$$

Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be the hierarchies of Figure 7.3(a) and (c), respectively. In Figure 7.3(h), we show the concatenation of $\mathcal{H}_1$ and $\mathcal{H}_2$ at level $\lambda = 2$.

7.3 Normalization of saliency maps

Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two hierarchical watersheds of (G, w). By the definition of hierarchical watersheds (Definition 1), we may affirm that $\mathcal{H}_1$ and $\mathcal{H}_2$ have the same number of partitions. Then, the range of both saliency maps $\Phi(\mathcal{H}_1)$ and $\Phi(\mathcal{H}_2)$ is equal to $\{0, \ldots, n-1\}$. Therefore, the weight of the contour of the k-th most important region of $\mathcal{H}_1$ is equal to the weight of the contour of the k-th most important region of $\mathcal{H}_2$. Furthermore, the number of regions of the k-th finest partition of $\mathcal{H}_1$ is equal to the number of regions of the k-th finest partition of $\mathcal{H}_2$. However, this might not be the case of two hierarchies computed from distinct methods. To ensure that the range of the saliency maps to be combined is compatible, we consider three normalization functions: N_1, N_2 and N_3.

Given any weight map $f \in \mathcal{F}$ and any edge u in E, the function N_1 assigns to u the number of values γ in the range of $\Phi(\mathcal{H})$ such that $f(u) \geq \gamma$:

$$N_1[f](u) = |\{f(v) \mid v \in E, f(u) > f(v)\}| \quad (7.5)$$

Let f_1 and f_2 be two saliency maps normalized by N_1. If the saliency maps f_1 and f_2 have distinct numbers of level-sets, those saliency maps can be re-normalized. For supremum and concatenation, we apply the function N_2 such that for any map f_i, for i in $\{1, 2\}$, and any edge u in E, we have:

$$N_2[f_i](u) = N_1[f_i](u) + (|\mathbb{W}_{max}| - |\mathbb{W}_i|) \quad (7.6)$$

where $|\mathbb{W}_i|$ is the cardinality of the range of f_i and $\mathbb{W}_{max}$ is the maximum cardinality in the set $\{|\mathbb{W}_1|, |\mathbb{W}_2|\}$.

Let f_1 and f_2 be two saliency maps. By applying the function N_2 to f_1 and f_2, we assure that the highest levels of the combination of f_1 and f_2 by supremum are not dominated by the levels of a single input hierarchy. Regarding combinations by concatenation, the function N_2 guarantees that the parameters of the combination actually indicate the number of levels taken from each hierarchy.

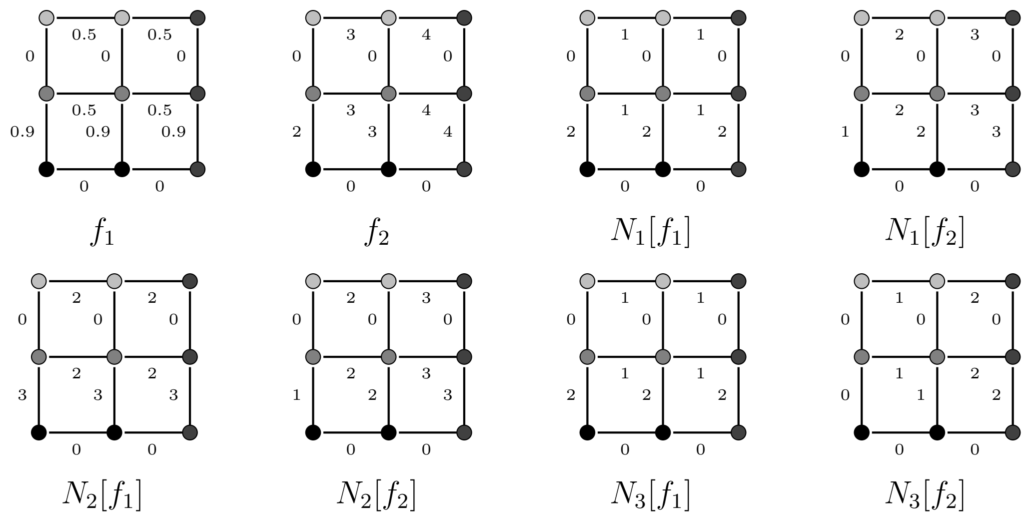


Figure 7.5: Two saliency maps f_1 and f_2 and their normalization by the functions N_1, N_2 and N_3.

In combinations by infimum and linear combinations, we discard the lowest levels of the hierarchy with the largest number of partitions by applying the normalization function N_3. For any map f_1, for $i \in \{1, 2\}$, and any edge $u \in E$, we have:

$$N_3[f_i](u) = max(N_1[f_i](u) - (|\mathbb{W}_i| - |\mathbb{W}_{min}|), 0) \tag{7.7}$$

where $|\mathbb{W}|_{min}$ is the minimum cardinality in the set $\{|\mathbb{W}_1|, |\mathbb{W}_2|\}$.

In Figure 7.5, we show two saliency maps f_1 and f_2, and the results of applying the normalization functions N_1, N_2 and N_3 to f_1 and f_2.

7.4 Visual inspection of combinations of hierarchies

In this section we present a visual inspection of combinations of hierarchical watersheds and of hierarchies based on non-increasing attributes. To assess the performance of each combination, we analyze the saliency maps and the partitions of the individual hierarchies and of their combinations.

To perform a visual inspection of combinations of hierarchies, we consider two gradient measures. The first measure corresponds to the Euclidean distance between neighboring pixels using the Lab color space. We call any gradient obtained through this measure a *Lab gradient*. As stated in [96], the euclidean distance in the Lab space captures better relevant boundaries according to human perception than the Euclidean distance in the RBG color space. As illustrated in Figure 7.6(b), the Lab gradient captures very small details of the original image, which can lead to oversegmentation. The advantages of this type of low-level gradient measure is its simple implementation and its capacity of preserving small and potentially important regions of an image.

The second method is the high-level *Structured Edge Detector (SED)* described in [29]. In [29], the authors trained an edge detector to infer if a given point in the image belongs to an edge using the information of a patch centered at this point. This edge detector produces high quality image contours. Figure 7.6 shows a color image and its SED gradient.

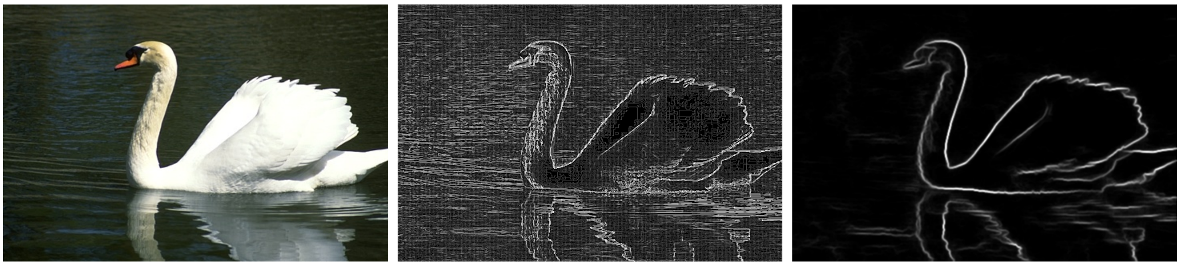


Figure 7.6: (a) original image I. (b) Lab gradient of I (c) SED gradient of I.

In the remainder of this section, we analyse combinations of hierarchies computed from Lab and SED gradients.

7.4.1 Combination of hierarchical watersheds by infimum

In this section, we illustrate combinations of hierarchical watersheds by infimum. We combine hierarchical watersheds computed from Lab and SED gradients. As the hierarchies to be combined are hierarchical watersheds of the same image gradient, there is no need for saliency map normalization. In Figure 7.7, we consider the area-based and dynamics-based hierarchical watersheds of three images computed from Lab gradient, and their combination by infimum. From each hierarchy, we extracted the segmentation which better describes the original image and which contains the least number of regions. For the first image, we search for a segmentation which separates the regions corresponding to the two people, the sea and the sky regions. For the second image, the optimal segmentation should segment the bear and the three layers of ice and vegetation. Concerning the third image, a good segmentation should segment the swan and the background. As a measure of quality, we consider the best hierarchy to be the hierarchy whose extracted segmentation contains the least number of regions.

For the three images illustrated in 7.7, the combinations of hierarchical watersheds

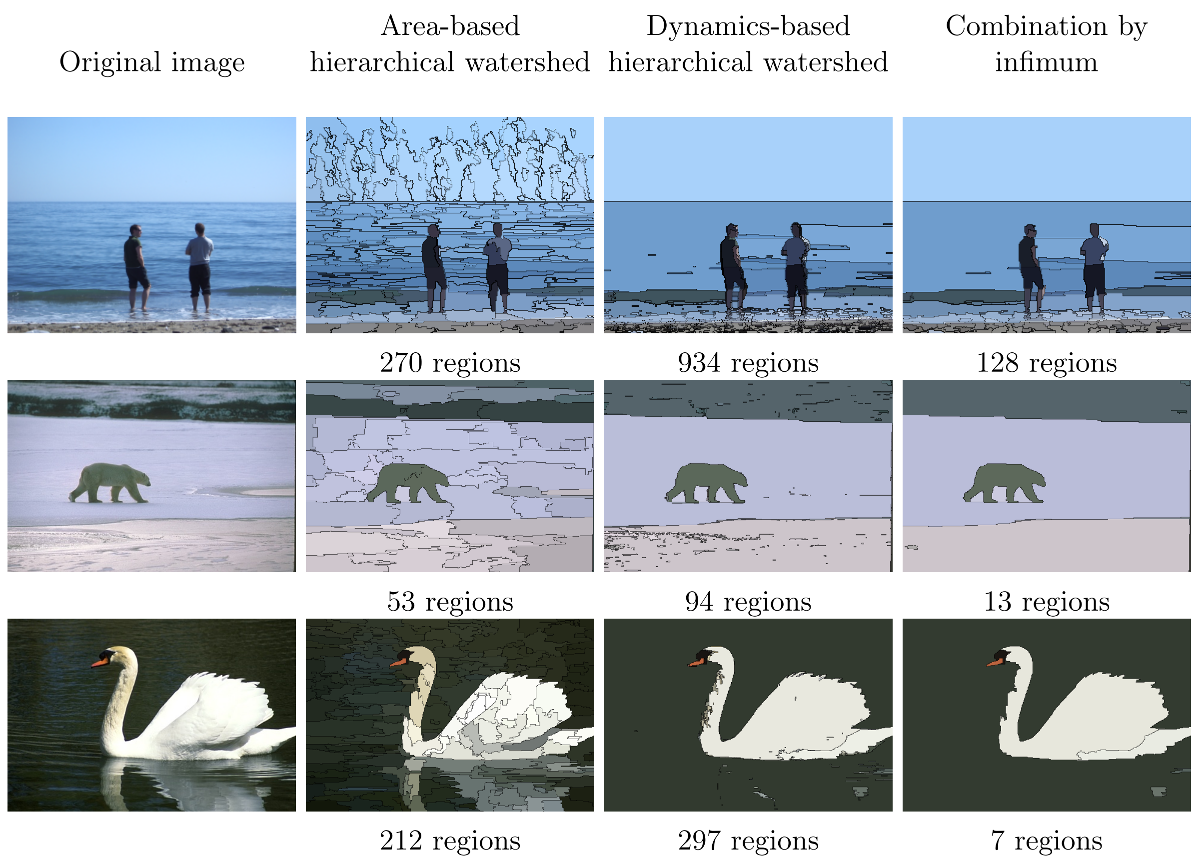


Figure 7.7: Original images, one segmentation extracted from each of the area-based and dynamics-based hierarchical watersheds, and a segmentation extracted from their combination by infimum. All hierarchies were computed from the Lab gradient. The optimal segmentation of each hierarchy was extracted according to the following criteria: First row: preservation of skyline and the two people. Second row: preservation of the bear, the space between its legs and the three layers of ice. Third row: preservation of the swan, keeping its body and beak in different regions.

computed from Lab gradient with infimum outperform the individual hierarchical watersheds by a great margin. However, this is not always the case for the combinations of area-based and dynamics-based hierarchical watersheds computed from the SED gradient (Figure 7.8). We can observe that, in the latter, the optimal segmentations of the image in the first and third rows are given by the dynamics-based hierarchical watersheds. Still, combinations perform better than the area-based hierarchical watersheds.

7.4.2 Combination of hierarchical watersheds by supremum

The visual inspection of combinations of hierarchical watersheds by supremum follows the same criteria as the combinations by infimum. In Figure 7.9 and Figure 7.10, we

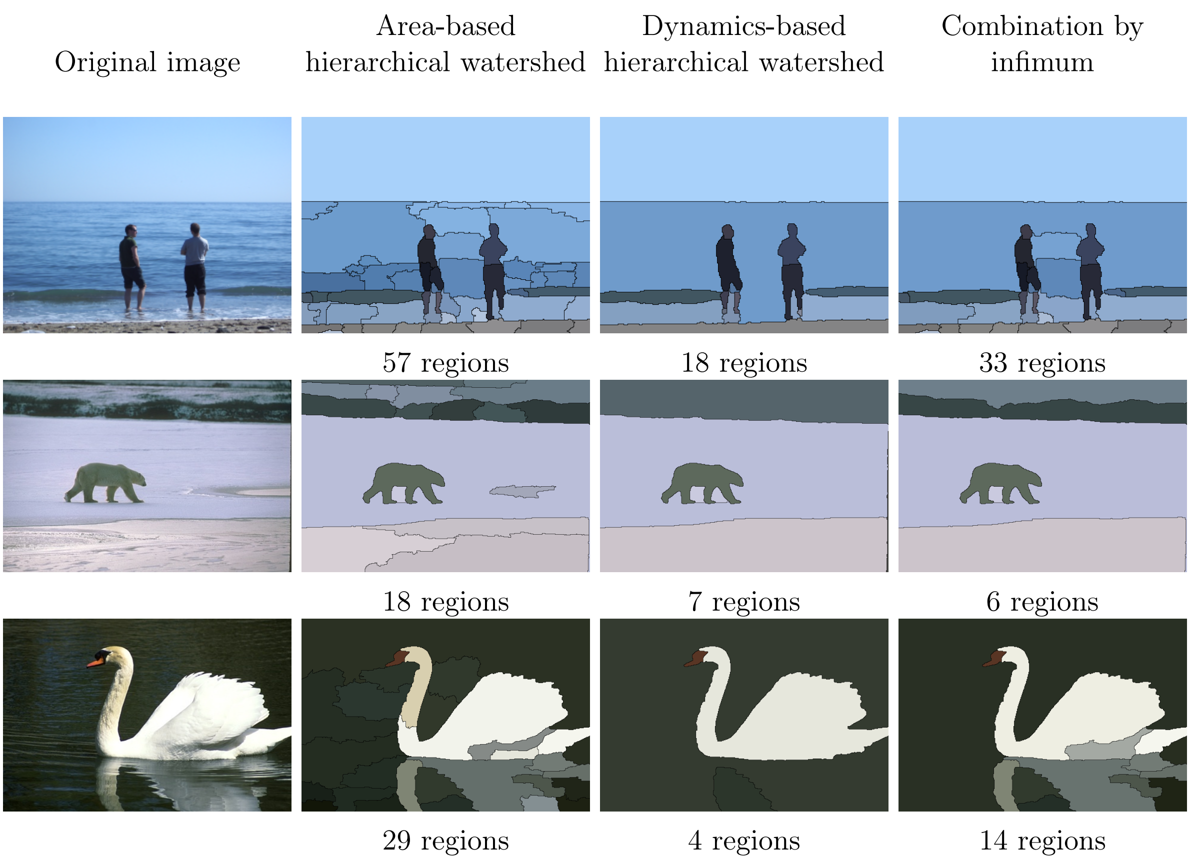


Figure 7.8: Original images, one segmentation extracted from each of the area-based and dynamics-based hierarchical watersheds, and a segmentation extracted from their combination by infimum. All hierarchies were computed from the SED gradient.

show combinations of hierarchical watersheds computed from Lab and SED gradients, respectively. We can observe that the combinations of hierarchical watersheds from Lab gradient do not outperform both individual area and dynamics based hierarchical watersheds. Indeed, the higher levels of the combinations with supremum preserve irrelevant boundaries of the area-based and dynamics-based hierarchical watersheds. Still, like the combinations with infimum, combinations of hierarchical watersheds computed from Lab and SED gradients outperform area-based hierarchical watersheds. Moreover, the combination by supremum of the third image of Figure 7.10 outperforms both hierarchical watersheds.

7.4.3 Combination of hierarchical watersheds by average

Combinations of area-based and dynamics-based hierarchical watersheds (computed using Lab gradient) by average are presented in Figure 7.11. Based on the same criteria

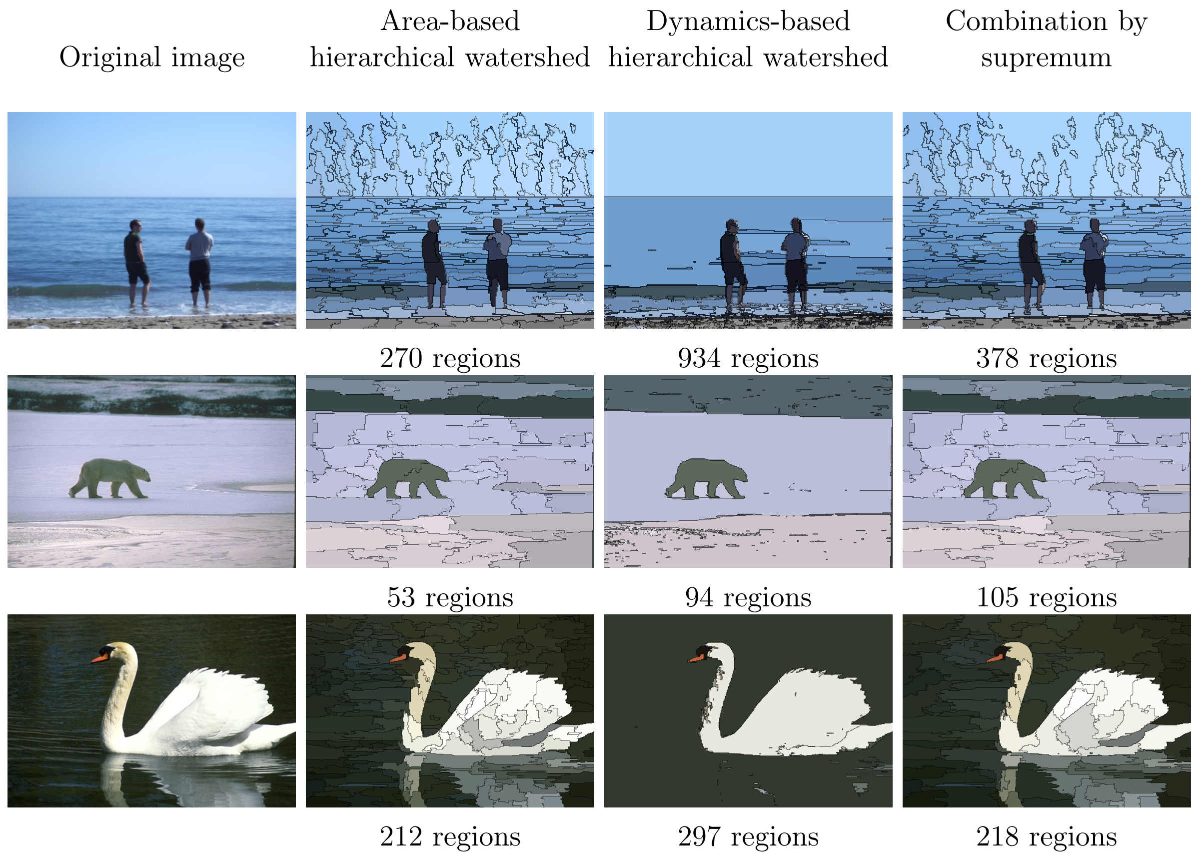


Figure 7.9: Original images, one segmentation extracted from each of the area-based and dynamics-based hierarchical watersheds, and a segmentation extracted from their combination by supremum. All hierarchies were computed from the Lab gradient.

upon which we inspected combinations by infimum and average, we can affirm that the three combinations with average illustrated in Figure 7.11 outperform the individual hierarchical watersheds. Moreover, it outperforms the combinations by infimum and supremum illustrated in figures 7.7 and 7.9.

Combinations of area-based and dynamics-based hierarchical watersheds (computed using SED gradient) by average are shown in Figure 7.12. Analogous to the combinations with infimum illustrated in Figure 7.8, the combinations with average of Figure 7.12 outperform the area-based hierarchical watersheds.

The inspection on combinations of hierarchical watersheds performed until now aimed to compare different combining functions with respect to a specific task. From the area and dynamics-based hierarchical watersheds and their combinations, we extracted the segmentation that better describes the main objects of each image and which contains the lowest number of regions. The objective was to bring to the fore the most important regions from the area and dynamics-based hierarchical watersheds through their combi-

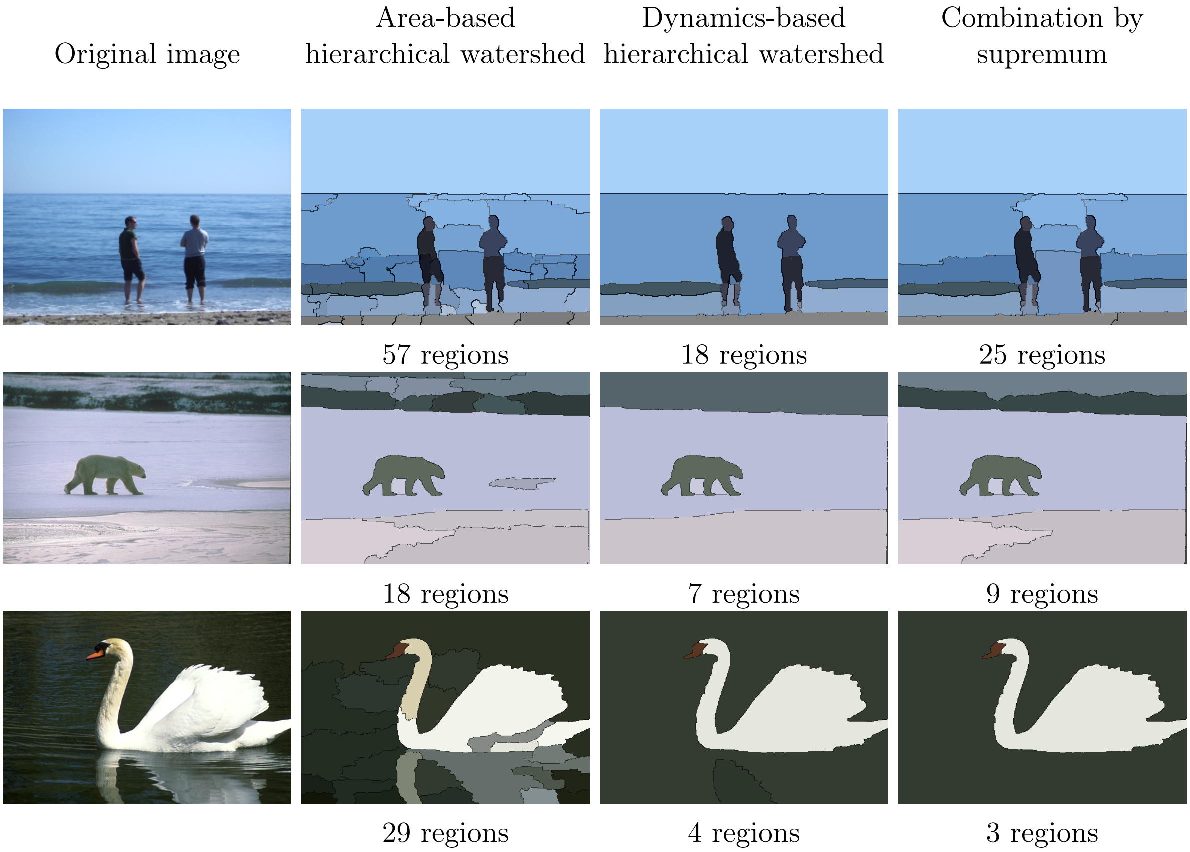


Figure 7.10: Original images, one segmentation extracted from each of the area-based and dynamics-based hierarchical watersheds, and a segmentation extracted from their combination by supremum. All hierarchies were computed from the SED gradient.

nations, which was the case for the most of combinations by infimum and average. As we will see later, our observations on combinations of hierarchical watersheds for this particular task do not generalize to combinations of hierarchical watersheds based on other increasing attributes.

7.4.4 Combination of hierarchical watersheds by concatenation

In general, the main large perceptual regions of an image I are better segmented at high levels of the volume-based hierarchical watershed of I when compared to the dynamics-based hierarchical watershed of I. In contrast, the opposite is true for the small details of the image I. Hence, we expect the concatenation of volume and dynamics-based hierarchical watersheds to highlight relevant regions of an image that are not relevant according to both criteria simultaneously.

In Figure 7.13, we shown an image I and three segmentations extracted from each

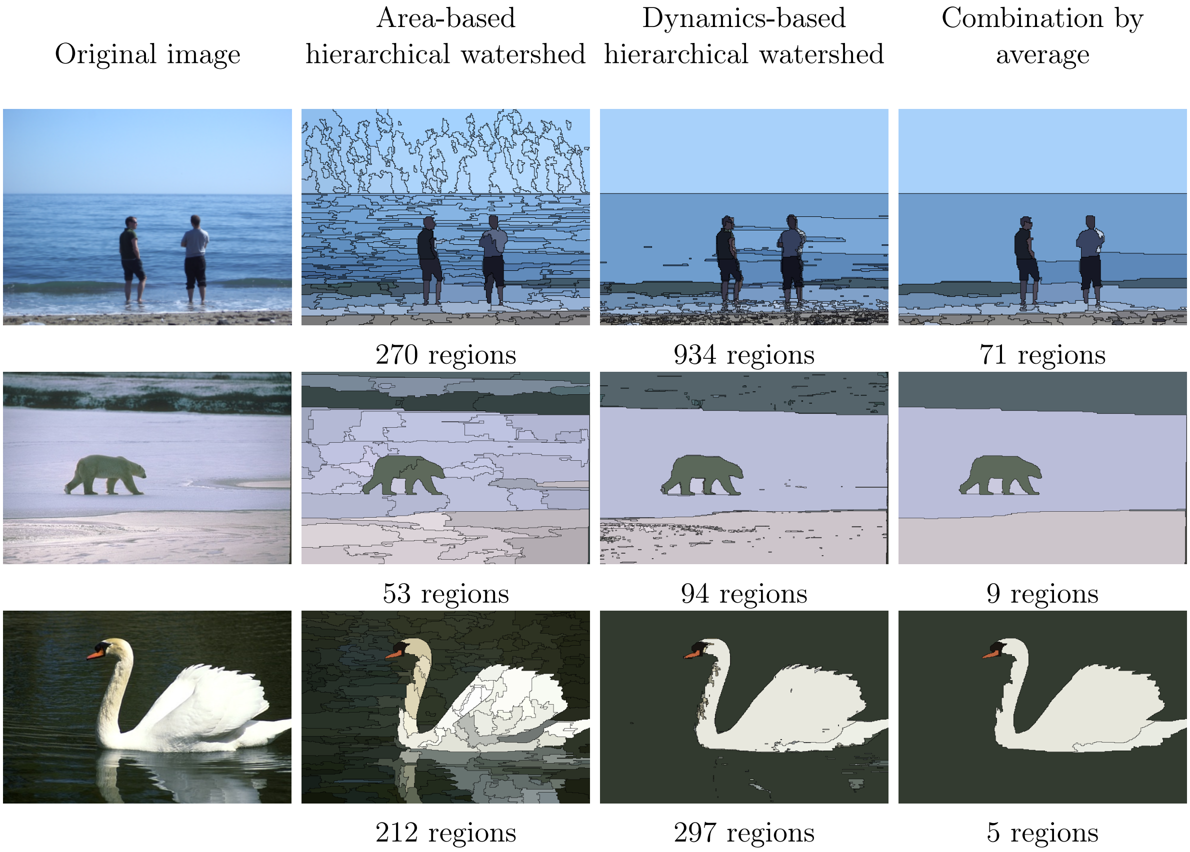


Figure 7.11: Original images, one segmentation extracted from each of the area-based and dynamics-based hierarchical watersheds, and a segmentation extracted from their combination by average. All hierarchies were computed from the Lab gradient.

of the following hierarchies: the volume-based hierarchical watershed computed from the SED gradient of I; the dynamics-based hierarchical watershed computed from the SED gradient of I; and the concatenation of the volume and dynamics-based hierarchical watersheds at level fourteen, *i.e.*, the fourteenth highest levels of volume with the lower levels of dynamics. The fourteenth highest levels of the concatenation of volume and dynamics are equal to the fourteenth highest levels of volume. From level fourteen downward, the concatenation brings forward some small regions highlighted by the dynamics, such as the symbol on the aircraft, which can be an important region depending on the application. Therefore, this type of combination can be useful when important regions of the image are not relevant according to the same criterion.

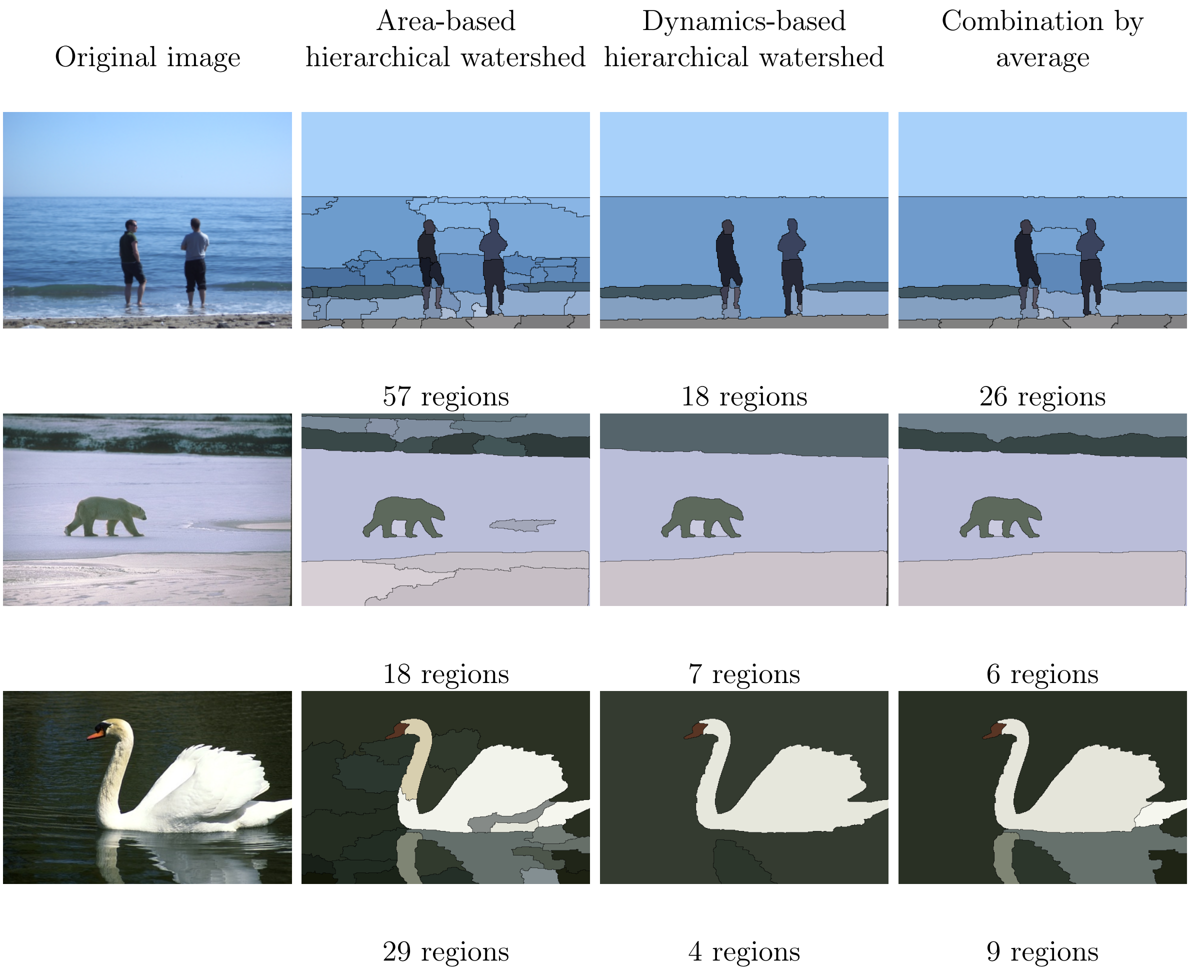


Figure 7.12: Original images, one segmentation extracted from each of the area-based and dynamics-based hierarchical watersheds, and a segmentation extracted from their combination by average. All hierarchies were computed from the SED gradient.

7.4.5 Combinations of hierarchical watersheds with hierarchies based on non-increasing attributes

There are cases where no hierarchical watershed based on an increasing criterion is able to highlight the desired regions of an image. For instance, the regions covering the low contrasted circular traffic signs in the two images of Figure 7.14 are not highlighted in none of the hierarchical watersheds of those two images. However, those circular regions are present at high levels of hierarchies based on circularity (see Section 2.7.3), which can be further used to improve hierarchical watersheds through combinations by supremum. In Figure 7.14, the hierarchies $\mathcal{H}_1$ and $\mathcal{H}_2$ are the hierarchical watersheds computed from the SED gradient of the images I_1 and I_2, respectively. The hierarchies $\mathcal{H}_1^c$ and $\mathcal{H}_2^c$ are the circularity based hierarchies computed from the saliency maps of the

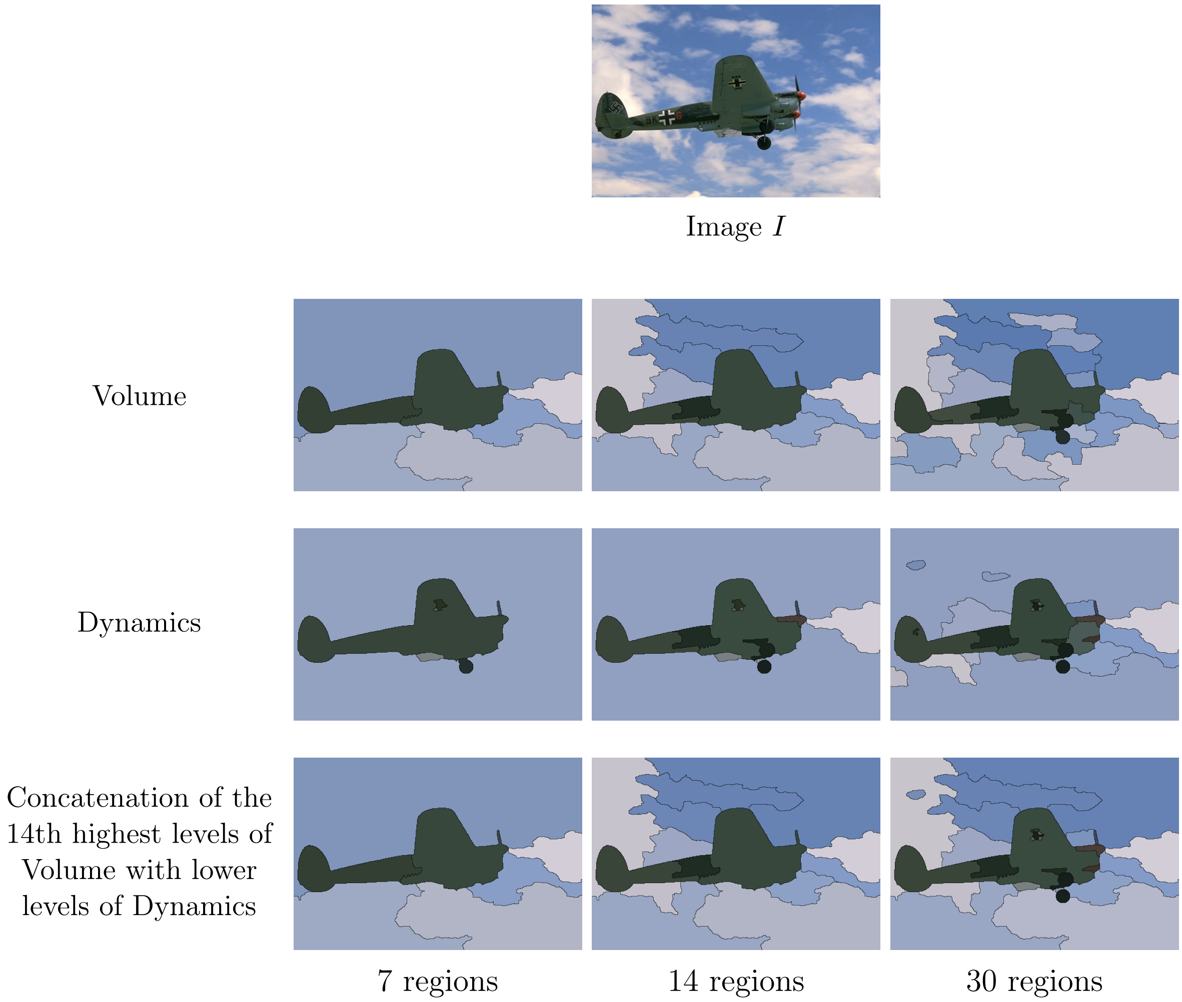


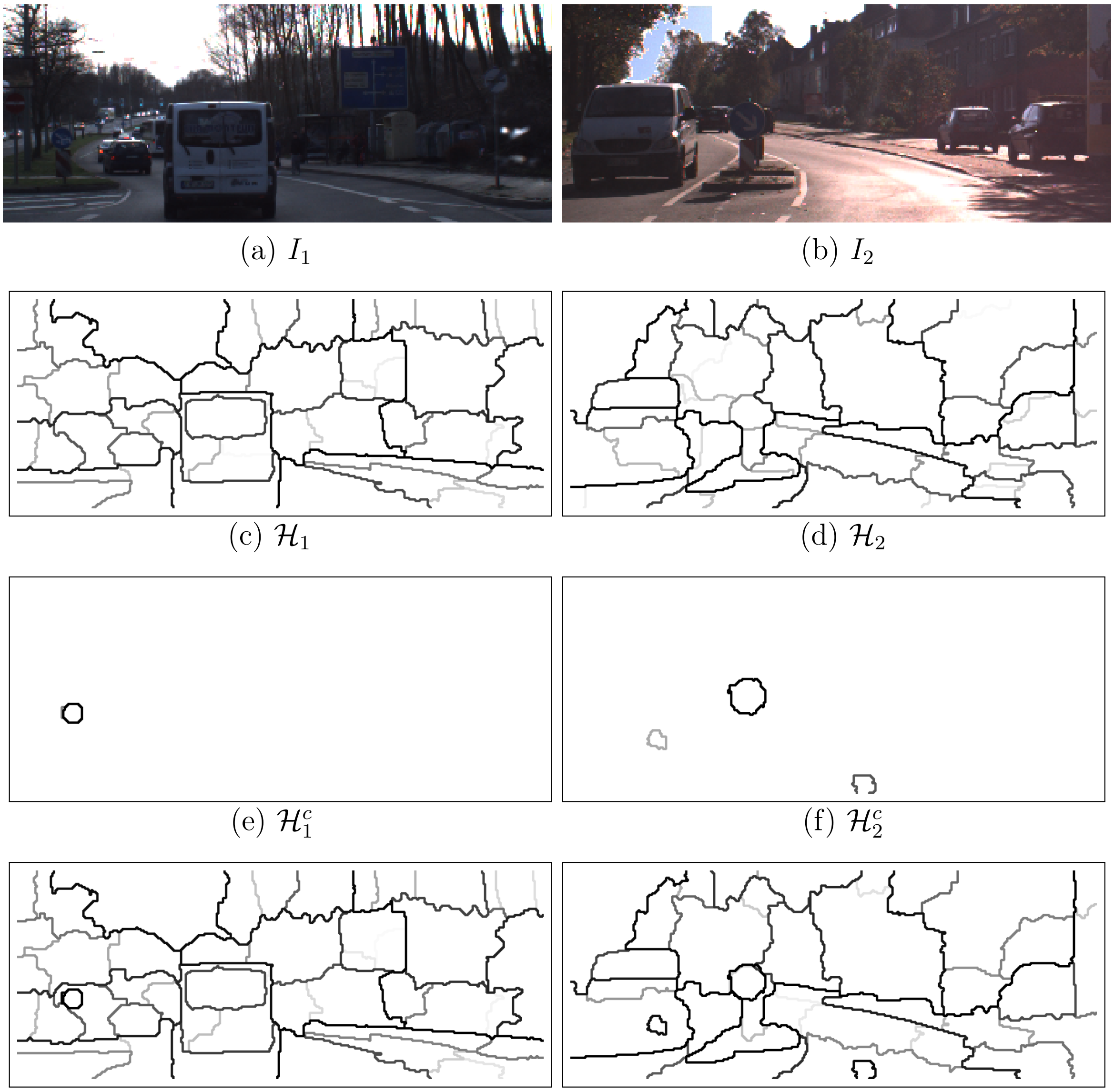
Figure 7.13: First row: original image I. Second row: three levels of volume (based hierarchical watershed) computed from the SED gradient of I. Third row: three levels of dynamics (based hierarchical watershed) computed from the SED gradient of I. Fourth row: three levels of the hierarchy resulting from the concatenation of the fourteenth highest levels of volume with lower levels of dynamics. The fourteenth highest levels of the concatenation of volume and dynamics are equal to the fourteenth highest levels of volume. From level 14 downward, the concatenation brings forward more regions highlighted by the dynamics, such as the symbol on the aircraft.

dynamics-based hierarchical watersheds of the images I_1 and I_2, respectively. In order to extract only the most circular regions from the circularity based hierarchies, we filtered out the regions with less than 100 pixels and whose circularity values were lower than 0.9 times the greatest circularity value in the hierarchy. The hierarchy $\mathcal{H}_1^{\curlyvee}$ (resp. $\mathcal{H}_2^{\curlyvee}$) is the combination of the hierarchies $\mathcal{H}_1$ and $\mathcal{H}_1^c$ (resp. $\mathcal{H}_2$ and $\mathcal{H}_2^c$) with supremum after normalization with N_2 (see Section 7.3). In both combinations, we are able to capture non-shape related information along with the low contrasted circular regions of each image.

In the example of Figure 7.14, the SED gradient of each of the images I_1 and I_2 succeeds at capturing the low-contrasted circular traffic signs, though the hierarchical watersheds computed from those gradients are not able to highlight those circular regions at high levels. In contrast, the SED gradient G of the image I of Figure 7.15 does not capture the low contrasted circular regions of I. In order to incorporate the circular regions of I into a hierarchical watershed computed from the high-quality gradient G, we combine the area-based hierarchical watershed $\mathcal{H}_a$ of Figure 7.15(c) with the circularity based hierarchy $\mathcal{H}_c$ of Figure 7.15(d) computed from the Lab gradient of I. More precisely, the hierarchy $\mathcal{H}_c$ is computed from the regions of the area-based hierarchical watershed of the Lab gradient of I. The hierarchy $\mathcal{H}^{\curlyvee}$ of Figure 7.15 is the supremum of $\mathcal{H}_a$ and $\mathcal{H}_c$. The hierarchy $\mathcal{H}^{\curlyvee}$ successfully combines the main large regions of I with the circular regions of I.

In Figure 7.16, we illustrate the use of combinations of hierarchies to highlight thin and elongated shapes of a retinal fundus picture. The dynamics-based hierarchical watershed $\mathcal{H}_d$ of Figure 7.16(b) highlights thick veins present in the image of Figure 7.16(a) but fails at capturing the thinner and low-contrasted veins. In turn, the perimeter based hierarchy $\mathcal{H}_p$ of Figure 7.16(c) successfully segments the thin veins in the center of the image I. However, the hierarchy $\mathcal{H}_d$ misses out the contours of a few high-contrasted veins. The combination of $\mathcal{H}_d$ and $\mathcal{H}_p$ by supremum, shown in Figure 7.16(d), balances the complementary information of both hierarchies.

In this section, we have shown that hierarchies based on non-increasing criteria can complement the segmentations of hierarchical watersheds without degrading the initial hierarchical watershed.



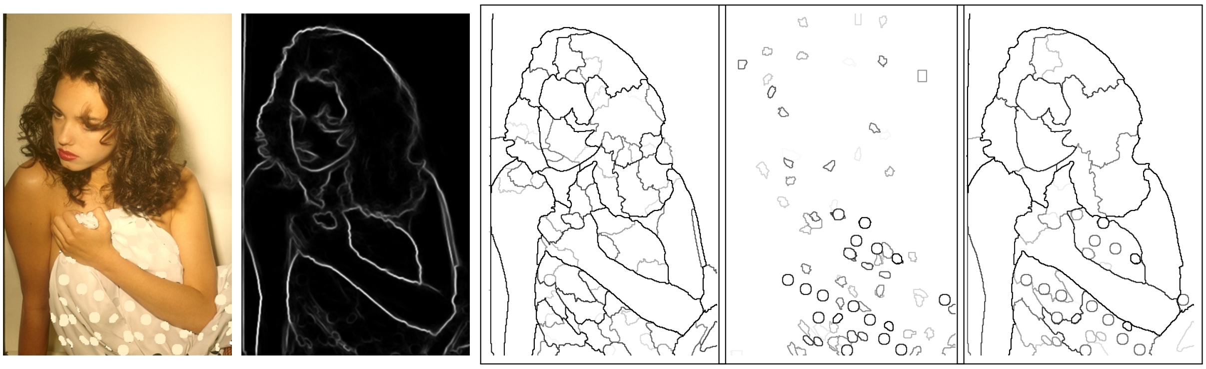
(a) I_1 (b) I_2

(c) $\mathcal{H}_1$ (d) $\mathcal{H}_2$

(e) $\mathcal{H}_1^c$ (f) $\mathcal{H}_2^c$

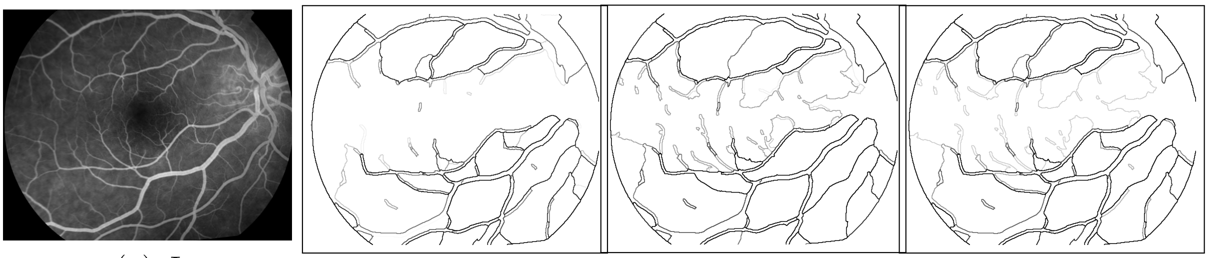
(g) $\mathcal{H}_1^\curlyvee$ (h) $\mathcal{H}_2^\curlyvee$

Figure 7.14: (a) and (b): original images from the German traffic sign dataset [49]. (c) $\mathcal{H}_1$: area-based hierarchical watershed computed of the SED gradient of I_1. (d) $\mathcal{H}_2$: area-based hierarchical watershed computed of the SED gradient of I_2. (e) $\mathcal{H}_1^c$: circularity based hierarchy computed from the regions of the dynamics-based hierarchical watershed of the SED gradient of I_1. (f) $\mathcal{H}_2^c$: circularity based hierarchy computed from the regions of the dynamics-based hierarchical watershed of the SED gradient of I_2. (g) $\mathcal{H}_1^\curlyvee$: combination of $\mathcal{H}_1$ and $\mathcal{H}_1^c$ by supremum. (h) $\mathcal{H}_2^\curlyvee$: combination of $\mathcal{H}_2$ and $\mathcal{H}_2^c$ by supremum.



(a) I (b) G (c) $\mathcal{H}_a$ (d) $\mathcal{H}_c$ (e) $\mathcal{H}^{\curlyvee}$

Figure 7.15: (a) Image I. (b) G: SED gradient of I. The low-contrasted circular regions of I do not appear in the SED gradient. (c) $\mathcal{H}_a$: Area-based hierarchical watershed of G. (d) $\mathcal{H}_c$: Circularity based hierarchy computed from the regions of the area-based hierarchical watershed of the Lab gradient of I. (e) $\mathcal{H}^{\curlyvee}$: Combination of $\mathcal{H}_a$ and $\mathcal{H}_c$ by supremum.



(a) I (b) $\mathcal{H}_d$ (c) $\mathcal{H}_p$ (d) $\mathcal{H}^{\curlyvee}$

Figure 7.16: (a) original image. (b) $\mathcal{H}_d$: dynamics-based hierarchical watershed of the SED gradient of I. (c) $\mathcal{H}_p$: perimeter based hierarchy computed from the regions of $\mathcal{H}_d$. (d) $\mathcal{H}^{\curlyvee}$: combinations of $\mathcal{H}_d$ and $\mathcal{H}_p$ by supremum.

7.5 Quantitative assessment of combinations of hierarchical watersheds

In this section, we present a quantitative evaluation of combinations of hierarchical watersheds. We first introduce the assessment methodology and the set-up of our experiments. Then, we present the evaluation of individual hierarchical watersheds. Subsequently, we present the evaluation of combinations of hierarchical watersheds in three parts: parameter-free combinations (infimum, supremum and average), unsupervised combinations by concatenation, and supervised linear combinations. Finally, we compare our best combination with a previous state-of-the-art approach.

7.5.1 Assessment methodology and set-up of experiments

We evaluate hierarchies based on the assessment framework presented in Chapter 6 (Section 6.5.1). This framework evaluates the possibility of extracting a *good* segmentation from a hierarchy with respect to a given ground-truth, the quality of the extracted segmentation being measured using the *Bidirectional Consistency Error* (BCE) [64]. In order to account for the hierarchical aspect of the representations, the score of a segmentation is measured against its level of fragmentation, *i.e.*, the ratio between the number of regions in the proposal segmentation compared to the number of regions in the ground-truth segmentation.

As established in Section 6.5.1, two ways of extracting segmentations from a hierarchy are considered:

- We compute the cut that maximizes the BCE score for each fragmentation level, leading to the Fragmentation-Optimal Cut score curve (FOC).
- We compute the BCE score of each partition of the hierarchy, leading to the Fragmentation-Horizontal Cut score curve (FHC).

The normalized area under those curves, denoted respectively by AUC-FOC and AUC-FHC, provides an overall performance summary over a large range of fragmentation levels. Since the importance of having high AUC-FOC and AUC-FHC scores varies according to the application, we consider the average of both scores to compare the performance of hierarchies. The average of AUC-FOC and AUC-FHC will be denoted here by AUC-FOHC.

In our experiments, we consider hierarchical watersheds based on the following increasing criteria: area [70, 98], dynamics [67], volume [98], topological height [91], number of minima, number of descendants, diagonal of bounding box [91], and number of parent nodes [79]. To shorten the notations, we denote those attributes by Area, Dyn, Vol, Height, Min, Desc, DBB and Parents. The hierarchical watersheds are computed from the SED gradient of the 200 test images of the Berkeley Segmentation Dataset (BSDS500) [63].

7.5.2 Baseline

Our baseline are the AUC-FOHC scores of individual hierarchical watersheds presented in Table 7.1. As we already discussed in Section 6.6, hierarchical watersheds based on the number of parent nodes outperform hierarchical watersheds based on the other aforementioned criteria. In the following sections, we investigate if there are combinations of hierarchical watersheds able to outperform the hierarchies based on number of parent nodes.

	Area	DBB	Dyn	Height	Desc	Min	Vol	Parent
AUC-FOC	0.603	0.604	0.541	0.560	0.604	0.609	0.617	0.620
AUC-FHC	0.423	0.423	0.480	0.493	0.425	0.453	0.465	0.505
AUC-FOHC	0.513	0.513	0.510	0.527	0.514	0.531	0.541	0.562

Table 7.1: AUC-FOC, AUC-FHC and AUC-FOHC scores of individual hierarchical watersheds computed over the test set of BSDS500.

7.5.3 Evaluation of parameter-free combinations

In tables 7.2, 7.3 and 7.4, we show the AUC-FOHC scores of the parameter-free combinations by infimum, supremum and average, respectively. We show in blue the combinations that outperform the initial hierarchies. The combinations that also outperform *Parent* are in blue and underlined.

The combinations by supremum improve the departing hierarchies in 13 over 28 combinations, in contrast to 10 combinations improved by infimum and average each. However, the contributions of combinations by infimum and average are higher than combinations by supremum. In particular, three combinations by infimum and five combinations by average perform better than *Parent*, which is not the case for any combination by supremum. Moreover, the best combinations are combinations by average: *Area* and *Height*, and *DBB* and *Height*.

Due to the promising results of combinations by average, we will investigate later in this section if there is a parameter better than 0.5 to perform linear combinations, *i.e.*, if there exist other linear combinations better than average.

7.5.4 Evaluation of unsupervised concatenation of hierarchies

In this section, we present the evaluation of concatenation of pairs of hierarchical watersheds.

To determine the parameter that should be used in the concatenation of each pair of hierarchical watersheds, we analyze the fragmentation curve of each hierarchy. For each pair of hierarchical watersheds, we check which one presents the highest AUC-FOC (resp. AUC-FHC) score for low and high fragmented segmentations. If one of the hierarchies

$\mathcal{H}_1\mathcal{H}_2$	Area	DBB	Dyn	Height	Desc	Min	Vol	Parent
Area	0.513	0.515	0.532	0.531	0.515	0.518	0.527	0.530
DBB		0.513	0.531	0.532	0.516	0.521	0.529	0.531
Dyn			0.510	0.516	0.535	0.546	0.551	0.560
Height				0.527	0.534	0.547	0.550	0.562
Desc					0.514	0.520	0.528	0.531
Min						0.531	0.540	0.542
Vol							0.541	0.555
Parent								0.562

Table 7.2: AUC-FOHC scores of combinations of pairs of hierarchical watersheds by supremum.

$\mathcal{H}_1\mathcal{H}_2$	Area	DBB	Dyn	Height	Desc	Min	Vol	Parent
Area	0.513	0.511	0.561	<u>0.564</u>	0.511	0.527	0.529	0.555
DBB		0.513	0.560	<u>0.564</u>	0.511	0.528	0.530	0.555
Dyn			0.510	0.522	0.560	0.557	0.542	0.541
Height				0.527	<u>0.564</u>	0.547	0.560	0.547
Desc					0.514	0.529	0.529	0.555
Min						0.531	0.533	0.554
Vol							0.541	0.553
Parent								0.562

Table 7.3: AUC-FOHC scores of combinations of pairs of hierarchical watersheds by infimum.

$\mathcal{H}_1\mathcal{H}_2$	Area	DBB	Dyn	Height	Desc	Min	Vol	Parent
Area	0.513	0.511	<u>0.566</u>	<u>0.568</u>	0.511	0.528	0.528	0.556
DBB		0.513	<u>0.566</u>	<u>0.569</u>	0.512	0.527	0.529	0.555
Dyn			0.510	0.521	<u>0.567</u>	<u>0.563</u>	0.550	0.550
Height				0.527	0.558	<u>0.563</u>	0.554	0.549
Desc					0.514	0.530	0.528	0.555
Min						0.531	0.534	0.553
Vol							0.541	0.556
Parent								0.562

Table 7.4: AUC-FOHC scores of combinations of pairs of hierarchical watersheds by average.

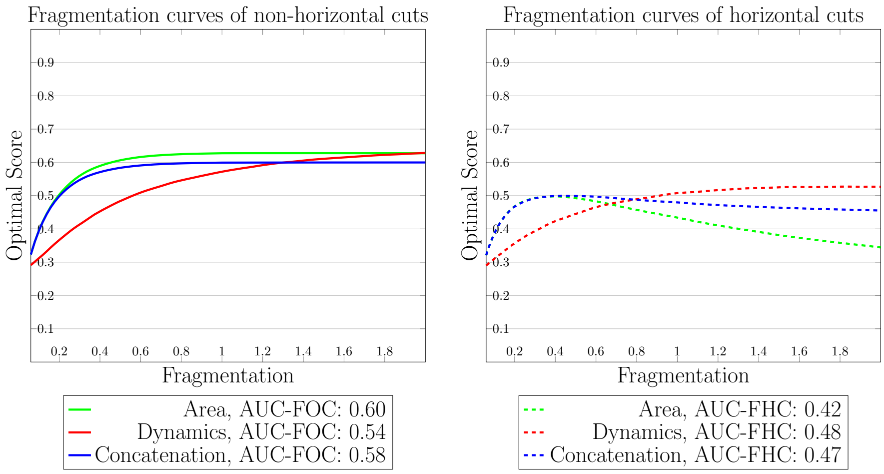


Figure 7.17: Fragmentation curves of non-horizontal and horizontal cuts of the concatenation of area and dynamics based hierarchical watersheds. The 10^{th} highest levels of area were concatenated to the lower level sets of the dynamics-based hierarchies.

presents the highest scores for both low and high fragmented segmentations, we do not expect to obtain better results from their concatenation. For example, in the curves of Figure 7.17, we see that, only for a fragmentation larger than 0 and smaller than approx. 0.65, area outperforms dynamics-based hierarchical watershed. Therefore, we conclude that high level sets of area, which are less fragmented, describe an image better than dynamics-based hierarchical watersheds, and the opposite is true for lower level sets. Hence, the parameters are tuned to concatenate high levels of area to the low levels of dynamics-based hierarchical watershed.

In general, the difference between the AUC-FHC and AUC-FOC scores is lower for the combinations with concatenation when compared to the initial hierarchies, which can be seen in the curves of Figure 7.17. This means that the segmentations extracted from the level sets of concatenations are closer to the optimal cuts for each fragmentation level. Moreover, nearly half of the concatenations tested here presented higher AUC-FOHC scores than the individual hierarchical watersheds (see Table 7.5).

7.5.5 Evaluation of supervised linear combinations

In this section, we present the evaluation of linear combinations of pairs of hierarchical watersheds using learned parameters.

For each pair of hierarchical watersheds, we determined the linear combination parameter α that optimizes the AUC-FOHC score on the 300 images of the training set of BSDS500. Then, we combined hierarchical watersheds computed on the images of the

$\mathcal{H}_2$	Dynamics					
$\mathcal{H}_1$	Area (10)	DBB (10)	Desc (12)	Min (14)	Vol (14)	Parent (18)
AUC-FOC	0.579	0.576	0.586	0.589	0.591	0.595
AUC-FHC	0.472	0.471	0.462	0.483	0.498	0.513
AUC-FHCO	0.525	0.523	0.526	0.536	0.545	0.554

$\mathcal{H}_2$	Height					
$\mathcal{H}_1$	Area (10)	DBB (9)	Desc (10)	Min (12)	Vol (12)	Parent (16)
AUC-FOC	0.579	0.570	0.580	0.582	0.585	0.591
AUC-FHC	0.472	0.472	0.473	0.485	0.500	0.513
AUC-FHCO	0.525	0.521	0.527	0.534	0.542	0.552

Table 7.5: AUC-FOC, AUC-FHC and AUC-FHCO scores of $\uplus_\lambda(\Phi(\mathcal{H}_1), \Phi(\mathcal{H}_2))$, where different values of λ were used for each concatenation. We concatenated the low levels of *Dynamics* and *Height* with the higher levels of *Area*, *DBB*, *Desc*, *Min*, *Vol* and *Parent*. The number of high levels taken from each hierarchy is shown in parenthesis. The AUC-FHCO scores in blue are the ones which are higher than the AUC-FHCO scores of individual $\mathcal{H}_1$ and $\mathcal{H}_2$ hierarchies.

test set of BSDS500 using the learned parameters.

The best-fitting parameter for each linear combination and their AUC-FOHC scores are shown in 7.6. The scores of the best combinations are in bold. Underlined are the combinations that outperform *Parents*, which account for half of the combinations. Interestingly, none of the combinations with *Parent* reached the highest score (0.569). Moreover, the optimal parameters of the other hierarchies when combined with *Parent* are nearly zero, which indicates that the resulting combinations are mostly influenced by *Parent*.

In Figure 7.18, we compare segmentations extracted from the hierarchical watersheds based on *Desc* and *Height*, and their linear combination using the learned parameters (see Table 7.6). The linear combination computed for this single image presents a higher AUC-FHC score than the individual hierarchies (0.604 *versus* 0.465 and 0.551) and a slightly higher AUC-FOC score (0.802 *versus* 0.801 and 0.708). Based on the AUC-FHC score, we expect this combination to have better horizontal cuts than the individual hierarchies. We can see that the segmentation extracted from the combination separates better the main regions in this image: sky, mountains and the two sea regions.

7.5.6 Comparison with other techniques

In order to have a more complete evaluation, we compare one of our best combinations with two high quality approaches: the Ultrametric Contour Map (UCM) [6] and the Multiscale Combinatorial Grouping (MCG) [81]. We include the other two assessment measures discussed in Chapter 6: Precision-Recall (PR) for boundaries and Object De-

$\mathcal{H}_1\mathcal{H}_2$	Area	DBB	Dyn	Height	Desc	Min	Vol	Parent
Area	- 0.513	$\alpha = 0.75$ 0.513	$\alpha = 0.60$ 0.568	$\alpha = 0.51$ **0.569**	$\alpha = 0$ 0.514	$\alpha = 0.11$ 0.531	$\alpha = 0$ 0.541	$\alpha = 0.03$ 0.563
DBB		- 0.514	$\alpha = 0.59$ 0.567	$\alpha = 0.52$ **0.569**	$\alpha = 0.18$ 0.514	$\alpha = 0.30$ 0.530	$\alpha = 0.0$ 0.541	$\alpha = 0.02$ 0.566
Dyn			- 0.510	$\alpha = 0.03$ 0.527	$\alpha = 0.38$ **0.569**	$\alpha = 0.51$ 0.564	$\alpha = 0.24$ 0.558	$\alpha = 0.11$ 0.566
Height				- 0.527	$\alpha = 0.42$ **0.569**	$\alpha = 0.51$ 0.560	$\alpha = 0.36$ 0.560	$\alpha = 0.21$ 0.566
Desc					- 0.514	$\alpha = 0.25$ 0.530	$\alpha = 0$ 0.541	$\alpha = 0.01$ 0.563
Min						- 0.531	$\alpha = 0.12$ 0.542	$\alpha = 0.01$ 0.562
Vol							- 0.541	$\alpha = 0.02$ 0.563
Parent								- 0.562

Table 7.6: Parameters α and AUC-FOHC scores of each linear combination $\boxplus_{(\alpha)}(\Phi(\mathcal{H}_1), \Phi(\mathcal{H}_2))$. The AUC-FOHC scores in bold are the highest scores achieved with linear combination of hierarchies.

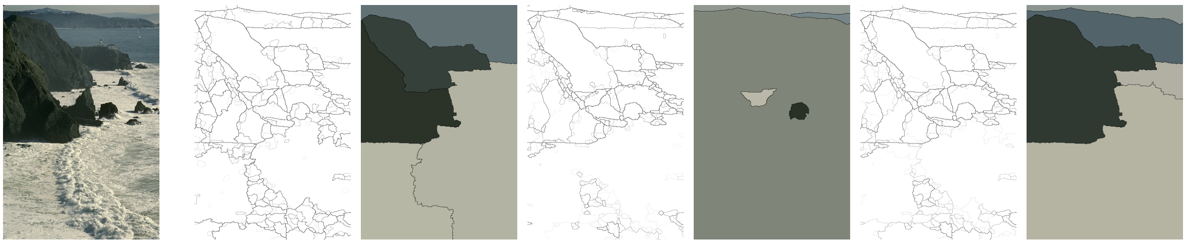


Figure 7.18: From left to right: original image, saliency map and the 5^{th} highest level set of three hierarchies: hierarchical watershed based on number of descendants, hierarchical watershed based on topological height, and their linear combination using learned parameters.

tection Measure.

The PR for boundaries score is assessed on BSDS500 and, as described in Chapter 6, it evaluates the matching between the boundaries of a given segmentation and the ground-truth segmentation. The PR curves are built from the precision and recall scores of the partitions of a hierarchy and are summed up in two F-measures: Optimal Dataset Scale (ODS) and Optimal Image Scale (OIS).

The Object Detection Measure is assessed over the Grabcut [16] and Weizman [1] datasets. Figure 7.19 shows the Object Detection Measure results for three pairs of background and foreground markers: Er-Er, Fr-Sk and Sk-Sk, in which Er, Fr and Sk stand for Erosion, Frame and Skeleton, respectively. The box plots show the quartile

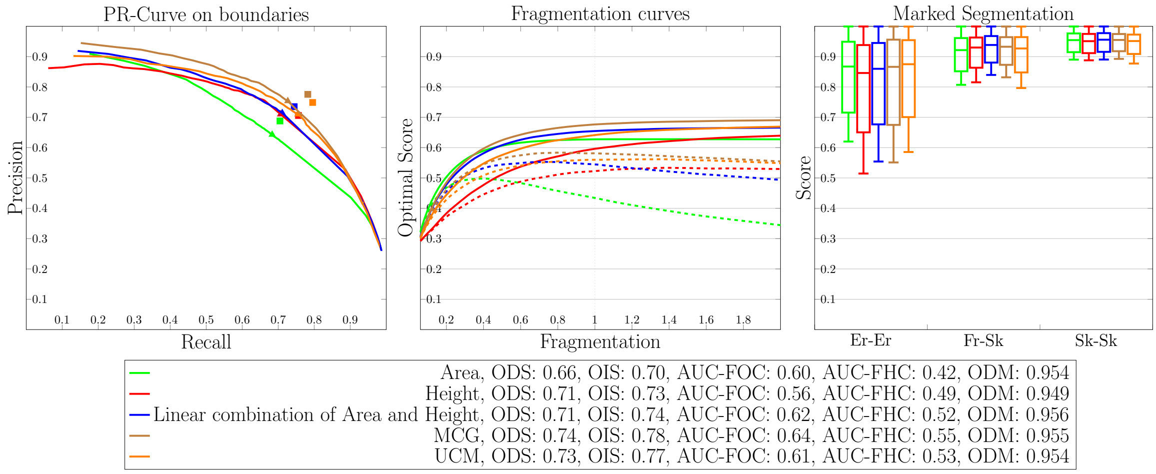


Figure 7.19: Comparison of our best linear combination with UCM [6] and MCG [81]: Precision-recall (PR) for boundaries, Fragmentation curves of non-horizontal cuts (plain curve) and horizontal cuts (dashed curve), and Object Detection Measure of three pairs of markers.

distribution of scores on both datasets. The median score of those three pairs of markers is denoted by ODM. Therefore, the best hierarchies in terms of Object Detection Measure correspond to the ones with highest ODM scores and most compressed box plots.

Our best combination does not achieve the PR and fragmentation scores presented by MCG and UCM, but it outperforms UCM in terms of fragmentation curves for non-horizontal cuts and presents competitive marked segmentation results compared to MCG and UCM.

7.6 Properties of combinations of hierarchical watersheds

Combining hierarchical watersheds through their saliency maps raises the question of whether the combination of hierarchies is closed for the set of hierarchical watersheds. More precisely, given any two hierarchical watersheds $\mathcal{H}_1$ and $\mathcal{H}_2$ of (G, w), is the combination of $\mathcal{H}_1$ and $\mathcal{H}_2$ with a given combining function also a hierarchical watershed of (G, w)?

In this section, we answer to this question for combinations of hierarchical watersheds by supremum, infimum, average and concatenation. We show that combinations with any of those functions do not result in hierarchical watersheds in general. However, in the particular case where we consider a unique ordering on the edges of (G, w), combinations of hierarchical watersheds with infimum have the noteworthy property of being flattened hierarchical watersheds, *i.e.*, hierarchies resulting from removing (and/or repeating) partitions of a departing hierarchical watershed (see Definition 21). For this particular case, we present a sufficient condition for a combining function to always output flattened hierarchical watersheds.

As stated in the following property, combinations of hierarchical watersheds by infimum, supremum, average and concatenation are not hierarchical watersheds nor flattened hierarchical watersheds in general.

Property 45. *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two hierarchical watersheds of (G, w). The combination of $\mathcal{H}_1$ and $\mathcal{H}_2$ by supremum (resp. average, concatenation and infimum) is not a flattened hierarchical watershed of (G, w) in general.*

Proof of Property 45. Let us consider the hierarchical watersheds $\mathcal{H}_1$ and $\mathcal{H}_2$ of the graph (G, w) and their combination $\mathcal{H}_\curlywedge$ by supremum depicted in Figure 7.20. The first pair of regions to be merged in $\mathcal{H}_\curlywedge$ are the minima B and C. However, there is no hierarchical watershed of (G, w) such that B and C are the first minima to be merged. By contradiction, let us assume that there is a sequence of minima $\mathcal{S}'$ of (G, w) such that, in the hierarchical watershed $\mathcal{H}'$ of (G, w) for $\mathcal{S}'$, B and C are the first minima to be merged. Then, we can infer that either B or C is the first minimum in the sequence $\mathcal{S}'$. However, if B were the first minima, then it would be merged to the minimum A because the weight of the edge linking A and B is 6, which is lower than the weight of the edge linking B and C. On the other hand, if C were the first minimum in the sequence $\mathcal{S}'$, then C would be first merged to D due to the same reason. Hence, there is no hierarchical watershed $\mathcal{H}'$ of (G, w) such that the partition $\{A, B \cup C, D, E\}$ of $\mathcal{H}_\curlywedge$ is a partition of $\mathcal{H}'$. Therefore, $\mathcal{H}_\curlywedge$ is not a flattened hierarchical watershed of (G, w). The same situation is found in the combinations by average and concatenation of the maps $\Phi(\mathcal{H}_1)$ and $\Phi(\mathcal{H}_2)$ shown in Figure 7.21. In both cases, the minima B and C are the first to be merged in the hierarchy. The counter example for infimum is presented in Figure 7.22. In Chapter 3, we introduced one-side increasing maps and their link with (flattened) hierarchical watersheds. In Figure 7.22, we show two maps f and g that are one-side increasing for the altitude orderings $\prec_1$ and $\prec_2$ for (G', w'), respectively. In fact, those are the only altitude orderings for (G', w'). The reader can verify that, by Property 22 on flattened hierarchical watersheds, the QFZ hierarchy of the combination of f and g with infimum is not a flattened hierarchical watershed of (G', w'). □

The counter-example of Figure 7.22 considers two maps which are one-side increasing (see Definition 18) for distinct altitude orderings for (G', w'). However, as established in

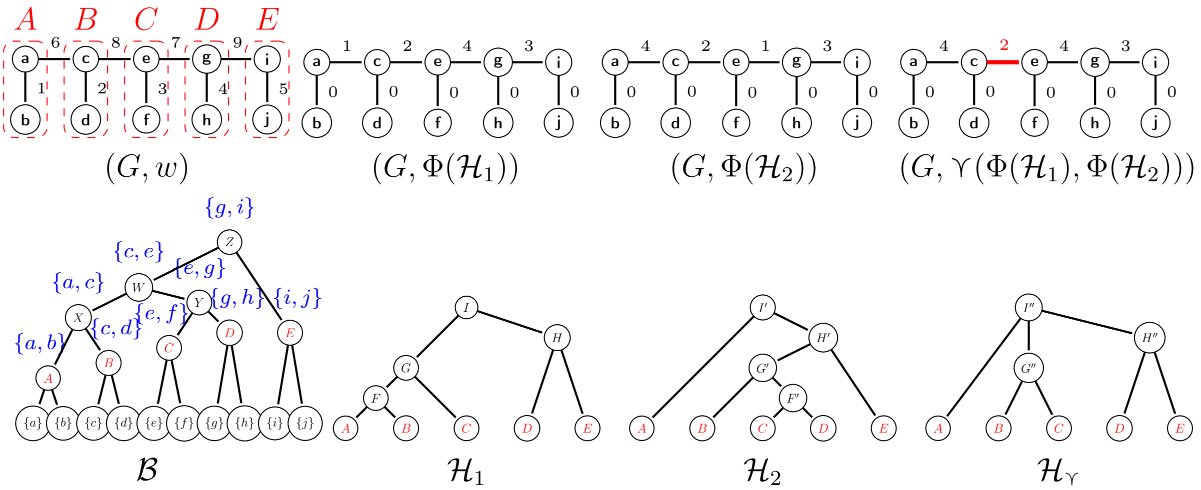


Figure 7.20: First line: a weighted graph (G, w) with five minima delimited by the dashed rectangles, the saliency maps $\Phi(\mathcal{H}_1)$ and $\Phi(\mathcal{H}_2)$ of two hierarchical watersheds of (G, w), and the combination $\curlyvee(\Phi(\mathcal{H}_1), \Phi(\mathcal{H}_2))$ of $\Phi(\mathcal{H}_1)$ and $\Phi(\mathcal{H}_2)$ with supremum. Second line: the unique binary partition hierarchy $\mathcal{B}$ of (G, w), the hierarchical watershed $\mathcal{H}_1$ of (G, w) for the sequence $\mathcal{S}_1 = (A, B, D, C, E)$, the hierarchical watershed $\mathcal{H}_2$ of (G, w) for the sequence $\mathcal{S}_2 = (D, C, E, A, B)$, and the resulting combinations of $\mathcal{H}_1$ and $\mathcal{H}_2$ with supremum: $\mathcal{H}_\curlyvee = \mathcal{QFZ}(G, \curlyvee(\Phi(\mathcal{H}_1), \Phi(\mathcal{H}_2))$.

Figure 7.21: Combination of the saliency maps $\Phi(\mathcal{H}_1)$ and $\Phi(\mathcal{H}_2)$ of Figure 7.20 with average and concatenation ($\lambda = 3$).

the following property, whose proof is presented in Appendix 8.4.1, if the input maps are one-side increasing for the same altitude ordering for (G, w), then their combination with infimum is a flattened hierarchical watershed of (G, w).

Property 46. *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two hierarchical watersheds of (G, w) and let $\prec$ be an altitude ordering for (G, w) such that both $\Phi(\mathcal{H}_1)$ and $\Phi(\mathcal{H}_2)$ are one-side increasing for $\prec$. Then the hierarchy $\mathcal{H}_\curlywedge = \mathcal{QFZ}(G, \curlywedge(\Phi(\mathcal{H}_1), \Phi(\mathcal{H}_2)))$ is a flattened hierarchical watershed of (G, w).*

Important remark: in the processing of graph-based image segmentation, it is common to consider a raster scanning of the edges of any input graph. Hence, when ties between edges of equal weights are solved deterministically, a unique altitude ordering $\prec$ for w is considered. Consequently, a unique binary partition hierarchy of (G, w) is used

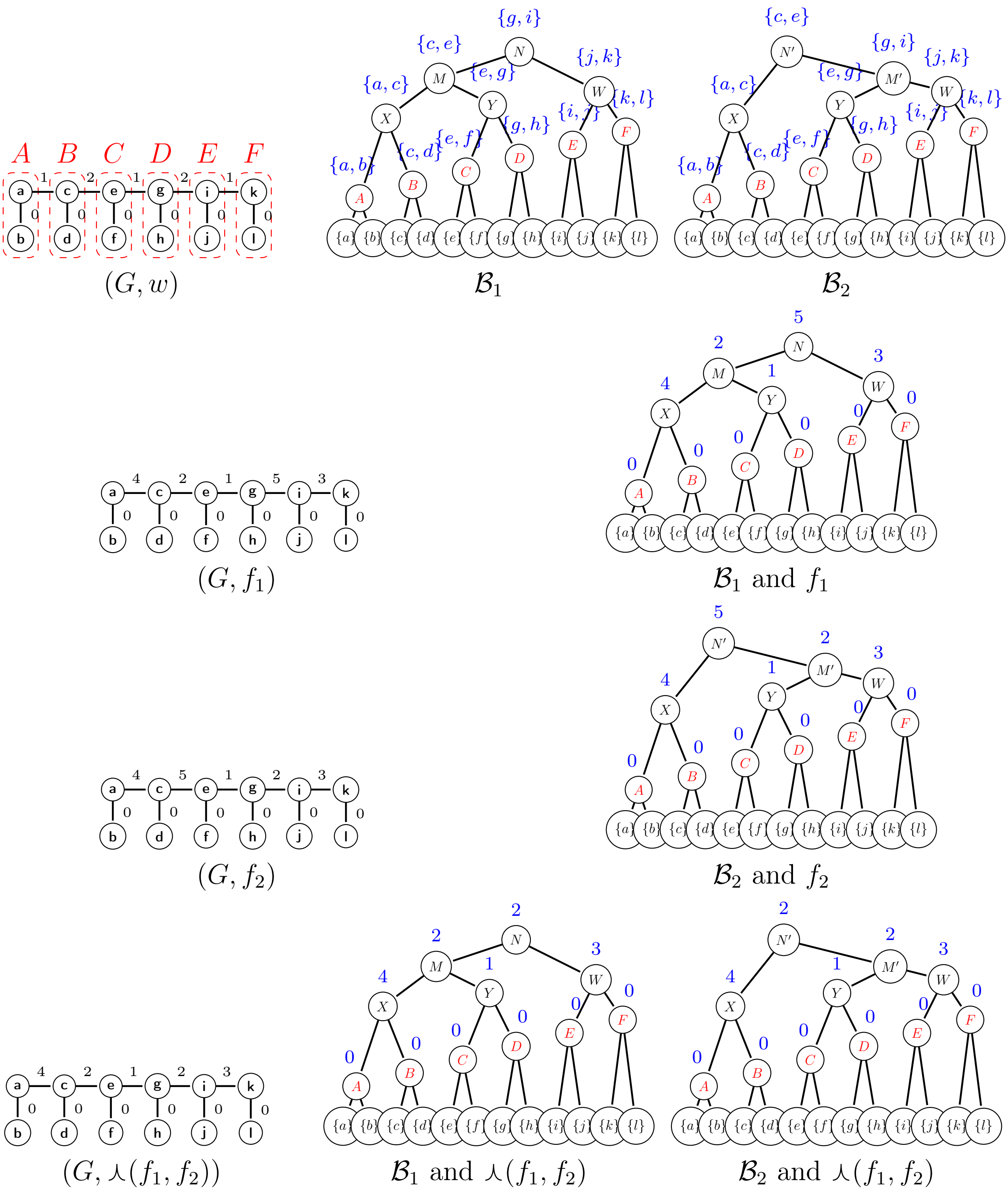


Figure 7.22: First line: a graph (G, w) and its two binary partition hierarchies $\mathcal{B}_1$ and $\mathcal{B}_2$. Let $\prec_1$ (resp. $\prec_2$) be the altitude orderings for (G, w) such that $\mathcal{B}_1$ (resp. $\mathcal{B}_2$) is the binary partition hierarchy of (G, w) for $\prec_1$ (resp. $\prec_2$). Second line: a map f_1 which is the saliency map of a hierarchical watershed of (G, w) and which is one-side increasing for $\prec_1$. Third line: a map f_2 which is the saliency map of a hierarchical watershed of (G, w) and which is one-side increasing for $\prec_2$. Fourth line: the combination of f_1 and f_2 with infimum, which is not one-side increasing for neither $\prec_1$ nor $\prec_2$.

by the algorithm.

In the following property, we introduce a sufficient condition for a combining function to always output flattened hierarchical watersheds.

Property 47. *Let C be a combining function, let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two hierarchical watersheds of (G, w) and let $\prec$ be and altitude ordering for (G, w) such that both $\Phi(\mathcal{H}_1)$ and $\Phi(\mathcal{H}_2)$ are one-side increasing for $\prec$. The combination of $\mathcal{H}_1$ and $\mathcal{H}_2$ with C is a flattened hierarchical watershed of (G, w) if $C(0, 0) = 0$ and if, for any a, b, c, d in $\{0, \ldots, n-1\}$, we have:*

1. *$C(a, b) = C(b, a)$; and*
2. *if $min(a, b) < min(c, d)$, then $C(a, b) < C(c, d)$; and*
3. *if $min(a, b) = min(c, d)$ and $max(a, b) < max(c, d)$, then $C(a, b) \leq C(c, d)$.*

The proof of Property 47 is detailed in Appendix 8.4.2.

By Property 47, we can derive other combining functions which always lead to flattened hierarchical watersheds, such as $C'(a, b) = min(a^m, b^m)$ and $C''(a, b) = min(a, b)^m$, for any $m \geq 1$, or the function stated in the following property, whose proof is given in Appendix 8.4.3.

Property 48. *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two hierarchical watersheds of (G, w). Let C be a combining function such that:*

$$C(x, y) = \begin{cases} 0 & if\ x{=}0\ and\ y{=}0 \\ \frac{x^m y^m}{x^m + y^m} & otherwise \end{cases} \tag{7.8}$$

for $m \geq n$. The combination of $\mathcal{H}_1$ and $\mathcal{H}_2$ with C is a flattened hierarchical watershed of (G, w).

7.7 Recognition of hierarchical watersheds applied to combinations of hierarchies

In Chapter 3, we proposed efficient algorithms to recognize (flattened) hierarchical watersheds. In this section, we apply those algorithms to the combinations of hierarchical watersheds evaluated in this chapter. By Property 48, combinations of hierarchical watersheds by infimum, supremum, linear combination and concatenation are not hierarchical watersheds in general. Indeed, by applying Algorithm 1 to the combinations of hierarchical watersheds with the aforementioned combining functions, we verified that the first condition of the definition of one-side increasing maps (Definition 18) is not satisfied by any combination. Hence, by Theorem 20, none of those combinations is a hierarchical watershed. In fact, combining hierarchies often act by simplifying the input hierarchies in the sense that, from a level i to a level $i+1$ of the resulting combination, zero or more than one pair of regions are merged, which suggests that combinations may result in flattened hierarchical watersheds.

To test if combinations of pairs of hierarchical watersheds by average, supremum and infimum are flattened hierarchical watersheds, we consider two cases during the computation of a hierarchical watershed: 1. when there are ties between edges of equal weights of a given input graph, an arbitrary choice is made; and 2. when there are ties between edges of equal weights of a given input graph, a deterministic choice is made. We applied Algorithm 2 to combinations of pairs of hierarchical watersheds considering each of those cases. The experiments were performed on the 200 images of the test set of BSDS500 [63]. For the case 1, the results are shown in Table 7.7. In each cell of Table 7.7, we present the number of combinations by average, by supremum and by infimum (among 200) that are flattened hierarchical watersheds. We can observe that the majority of the combinations with average and supremum are flattened hierarchical watersheds, which is not the case for combinations with infimum.

Given any two hierarchical watersheds $\mathcal{H}_1$ and $\mathcal{H}_2$ computed from the same image gradient g and based on distinct attributes, we cannot guarantee that the saliency maps of $\mathcal{H}_1$ and $\mathcal{H}_2$ are one-side increasing for a same altitude ordering for g. However, if that were the case, the resulting combination of $\mathcal{H}_1$ and $\mathcal{H}_2$ with infimum would be a flattened hierarchical watershed, as established in Property 47. By applying Algorithm 2 to combinations of saliency maps obtained by the deterministic algorithm (case 2), we observed that all combinations with infimum are flattened hierarchical watersheds as expected. Interestingly, this was also the case for the combinations with average. Regarding the combinations with supremum, among all 5600 combinations, only one combination with volume and diagonal of bounding box, and three combinations with volume and height were not flattened hierarchical watersheds.

Our experimental results suggest that most of the combinations of hierarchical watersheds are "approximations" of flattened hierarchical watersheds in the sense that, by swapping the weight of a few edges in the combinations of saliency maps, we could obtain a flattened hierarchical watershed.

$\mathcal{H}_1 \mid \mathcal{H}_2$	Area	DBB	Dyn	Height	Desc	Min	Vol	Parent
Area	-	138	125	135	152	59	113	79
	-	200	194	194	200	183	198	182
	-	119	113	124	127	47	60	62
DBB	-	-	127	134	136	62	119	82
	-	-	195	197	200	184	198	183
	-	-	115	122	113	48	102	65
Dyn	-	-	-	117	124	104	126	105
	-	-	-	195	195	189	196	192
	-	-	-	91	111	96	112	100
Height	-	-	-	-	134	108	128	110
	-	-	-	-	195	185	194	186
	-	-	-	-	123	99	106	97
Desc	-	-	-	-	-	63	114	83
	-	-	-	-	-	185	199	180
	-	-	-	-	-	52	98	65
Min	-	-	-	-	-	-	66	171
	-	-	-	-	-	-	179	199
	-	-	-	-	-	-	53	158
Vol	-	-	-	-	-	-	-	80
	-	-	-	-	-	-	-	177
	-	-	-	-	-	-	-	66
Parent	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-

Table 7.7: In each cell, we show the number of combinations of hierarchical watersheds by average (red), supremum (blue) and infimum (black) among 200 that are flattened hierarchical watersheds.

7.8 Watersheding of combinations of hierarchical watersheds

From the results of the previous section, we concluded that combinations of hierarchical watersheds are approximations of flattened hierarchical watersheds. We can go further into this hypothesis by computing the watersheding of combinations of hierarchical watersheds in order to answer to the following question: are there hierarchical watersheds which perform as well (or better) than the combinations of hierarchical watersheds studied so far? We answered to this question by applying the watersheding operator (see Chapter 4) to the combinations by infimum, supremum and average studied here.

In tables 7.8, 7.9 and 7.10, we present the AUC-FOHC scores of the watersheding of combinations of hierarchical watersheds by supremum, infimum and average, respectively. The results that outperform the individual hierarchies are in blue and the ones that outperform *Parent* are underlined. By comparing those tables with the scores of combinations by supremum, infimum and average shown respectively in tables 7.2, 7.3 and 7.4, we can observe that the majority of the scores of the watersheding of the com-

$\mathcal{H}_1\mathcal{H}_2$	Area	DBB	Dyn	Height	Desc	Min	Vol	Parent
Area	0.513	0.514	0.532	0.531	0.515	0.521	0.527	0.532
DBB	-	0.513	0.532	0.533	0.515	0.523	0.530	0.533
Dyn	-	-	0.510	0.517	0.533	0.546	0.553	0.561
Height	-	-	-	0.527	0.533	0.547	0.552	0.563
Desc	-	-	-	-	0.514	0.522	0.529	0.532
Min	-	-	-	-	-	0.531	0.540	0.543
Vol	-	-	-	-	-	-	0.541	0.556
Parent	-	-	-	-	-	-	-	0.562

Table 7.8: AUC-FOHC scores of the watersheding of combinations of hierarchical watersheds by supremum.

$\mathcal{H}_1\mathcal{H}_2$	Area	DBB	Dyn	Height	Desc	Min	Vol	Parent
Area	0.513	0.513	0.561	0.564	0.513	0.530	0.530	0.557
DBB	-	0.513	0.561	0.564	0.513	0.531	0.531	0.557
Dyn	-	-	0.510	0.523	0.561	0.558	0.543	0.543
Height	-	-	-	0.527	0.564	0.560	0.548	0.549
Desc	-	-	-	-	0.514	0.532	0.531	0.557
Min	-	-	-	-	-	0.531	0.538	0.556
Vol	-	-	-	-	-	-	0.541	0.556
Parent	-	-	-	-	-	-	-	0.562

Table 7.9: AUC-FOHC scores of the watersheding of combinations of hierarchical watersheds by infimum.

binations are at least as good as the scores of the combinations. Furthermore, we were able to outperform *Parent* in one more combination by supremum.

Hence, by coupling the framework of combination of hierarchical watersheds with the watersheding operator, we have an efficient tool to improve hierarchical watersheds with the advantage of preserving their mathematical properties.

$\mathcal{H}_1\mathcal{H}_2$	Area	DBB	Dyn	Height	Desc	Min	Vol	Parent
Area	0.513	0.513	0.567	0.569	0.514	0.532	0.530	0.559
DBB	-	0.513	0.567	0.570	0.513	0.531	0.531	0.556
Dyn	-	-	0.510	0.522	0.568	0.565	0.552	0.551
Height	-	-	-	0.527	0.569	0.564	0.555	0.552
Desc	-	-	-	-	0.514	0.533	0.531	0.558
Min	-	-	-	-	-	0.531	0.540	0.555
Vol	-	-	-	-	-	-	0.541	0.559
Parent	-	-	-	-	-	-	-	0.562

Table 7.10: AUC-FOHC scores of the watersheding of combinations of hierarchical watersheds by average.

7.9 Conclusion

We studied theoretical and practical aspects of a framework of combination of hierarchies. Through a visual inspection, we showed that this framework can be used to combine the strength of hierarchies computed from distinct criteria such as area and circularity. Then, we presented a quantitative evaluation of combinations (of pairs of hierarchical watersheds) over a large image dataset. We concluded that combining hierarchical watersheds is a valuable method to outperform individual hierarchical watersheds. Using the results of Chapter 3 on the recognition of hierarchical watersheds, we concluded that combinations of hierarchical watersheds by supremum, infimum, linear combination and concatenation are not necessarily hierarchical watersheds. However, those combinations can be approximated into hierarchical watersheds using the watersheding operator presented in Chapter 4. Our results showed that the watersheding of combinations perform even better than most of the combinations.

From the results of the watersheding operator applied to combinations of hierarchical watersheds, a natural question arises: can we obtain the hierarchical watersheds resulting from the watersheding operator without passing by the combination process? Or, can we learn an attribute that could be used to directly compute the results of the watersheding operator? A potential departing point to answering those questions would be to investigate if there is any combination of the attributes used here that corresponds to the attributes used to compute the hierarchies resulting from the watersheding.

Chapter 8

Conclusion

The general purpose of this thesis was to better understand the potential of hierarchical watersheds and their properties, which can be used on the development of new data processing tools. To conclude this manuscript, we review our methodological, experimental, theoretical and algorithmic contributions, and we discuss future work perspectives.

Methodological contributions

Our methodological contributions comprise the proposed evaluation framework (Chapter 6) of hierarchies of partitions and the watersheding operator (Chapter 4).

Numerous hierarchical segmentation methods have been proposed in the last few decades. Each of those methods are either general or aimed at specific applications, such as natural image segmentation, single object detection, and image compression and simplification. To provide a mean to compare different hierarchical segmentation methods, we proposed an evaluation framework to empirically assess those methods. This framework allowed us to measure the impact of image gradient choice and of small regions filtering in the performance of hierarchies of segmentations. It also played an important role in our experiments, namely on the evaluation of combinations of hierarchical watersheds and on the the learning of optimal parameters to combine hierarchies. Moreover, this framework allowed us to discover a new regional attribute called *Number of Parent Nodes* that, when used to compute hierarchical watersheds, outperform the commonly used area, dynamics and volume attributes.

Inspired by the mathematical properties of hierarchical watersheds and by the quantitative evaluation of these hierarchical, we proposed the *watersheding* operator. This idempotent operator converts any hierarchy of segmentations into a hierarchical watershed of a given (edge-weighted) graph. Among the potential applications of the watersheding operator, we highlighted (1) the computation of hierarchical watersheds based on non-increasing attributes and (2) the refinement of coarse hierarchies, *i.e.*, hierarchies that do not include fine regions of the input image.

The properties of the watersheding operator and the visual inspection of a few results suggest that the watersheding of a hierarchy is an "approximation" of this hierarchy. In other words, that merging order of the regions in an input hierarchy are preserved "as much as possible" in the watersheding of this hierarchy. As future work, we hope to formalize this assumption by studying whether there is any relevant objective function optimized by the watersheding operator.

Theoretical contributions

The well established properties of hierarchical watersheds, in particular the link between watersheds, minimum spanning forests and binary partition hierarchies, provided us a solid basis to investigate theoretical aspects of hierarchical watersheds. Our main theoretical contributions include a novel characterization of hierarchical watersheds (Chapter 3), the study of the properties of the watersheding operator (Chapter 4), and the introduction of the notion of probability of a hierarchical watershed (Chapter 5).

As discussed in Chapter 2, hierarchical watersheds are often obtained from a sequence (total ordering) of minima of a graph. Given a sequence of minima $\mathcal{S}$ and the watershed segmentation $\mathbf{P}$ of a graph, we obtain the hierarchical watershed for $\mathcal{S}$ by iteratively merging the regions of $\mathbf{P}$ in the order defined by $\mathcal{S}$. A question that emerged from this definition is whether any sequential mergings of the regions of $\mathbf{P}$ lead to a hierarchical watershed. In other words, given a hierarchy obtained by merging the regions of a watershed segmentation of a graph, can we always find a sequence of minima that lead to this given hierarchy? To answer to this question, we provided a simple characterization of hierarchies of watersheds, which was also the basis for defining the watersheding operator. Based on this proposed characterization, we showed that not all hierarchies whose first level is a watershed segmentation are hierarchical watersheds. As future work, we want to investigate if there are any state-of-the-art hierarchies that, with the help of the watersheding operator, can be approximated into hierarchical watersheds without performance degradation. If that were the case, we could use those state-of-the-art hierarchies to learn optimal attributes to compute hierarchical watersheds.

The definition of hierarchical watersheds obtained from sequences of minima (of a graph) has also incited the following questions: can a hierarchy be the hierarchical watershed (of a given graph) for more than one sequence of minima? And, in the affirmative case, are the hierarchical watershed of a given graph obtained from different numbers of sequences of minima? As answered in Chapter 5, when the input graph has pairwise distinct weights and at least three minima, the answer to both questions is positive: there are at least two hierarchies which can be obtained from a different number of sequences of minima. This finding led us to define and characterize the most and least probable hierarchical watersheds of a given graph. Potential directions of this work include (1) the extension of those theoretical results to arbitrary graphs and (2) the study of the probability of hierarchical watersheds based on the well-known attributes such as area, dynamics and volume. For instance, we could aim at answering questions like "are dynamic-based hierarchical watersheds more probable than area-based hierarchical watersheds?" or "what does the probability of the best hierarchical watersheds of a given image gradient tells us about the quality of this gradient?".

Experimental contributions

All experiments performed in this research were oriented to natural image segmentation (Chapter 7). We first explored a method to combine hierarchical segmentations through their saliency maps using functions as infimum, supremum and average. Based on the visual inspection of a few combinations and on the quantitative results obtained from our evaluation framework, we concluded that combining the saliency maps of hierarchical watersheds is a simple and valuable way to improve hierarchies for the general segmentation problem. For applications aiming to highlight objects with specific geometric properties, combinations of hierarchical watersheds with hierarchies based on non-increasing shape attributes are also a viable choice.

Thank to our proposed characterization of hierarchical watersheds, we concluded that combinations of hierarchical watersheds are not necessarily hierarchical watersheds. However, our experimental results proved that the majority of the tested combinations are *flattened hierarchical watersheds*, *i.e.*, hierarchies composed of a subset of segmentations of a hierarchical watershed. Moreover, by applying the watersheding operator to combinations of hierarchical watersheds, we obtained hierarchical watersheds which outperformed the given combinations.

For future work, an interesting line of research would be to combine saliency maps using the watersheding operator. For instance, given two saliency maps s_1 and s_2, we can view the watersheding of s_1 (resp. s_2) with respect to s_2 (resp. s_1) as a combination of s_1 and s_2. Hence, instead of computing the watersheding of a saliency map with respect to an image gradient, we would compute the watersheding of a saliency map with respect to another saliency map. In particular, combinations with the watersheding operator make sense when the boundaries of the input saliency maps overlap, which can be the case of hierarchical watersheds computed from the same graph. Hence, we could investigate the advantages and disadvantages of performing combinations using the watersheding operator instead of using the combination framework evaluated in Chapter 7.

Algorithmic contributions

The applicability of our theoretical contributions stems from the efficient algorithms derived from those results. Thank to the link between hierarchical watersheds and binary partition hierarchies, we designed quasi-linear algorithms to recognize hierarchical watersheds, to obtain the watersheding of a hierarchy and to compute the probability of a hierarchical watershed.

References

[1] S. Alpert, M. Galun, A. Brandt, and R. Basri. Image segmentation by probabilistic bottom-up aggregation and cue integration. *IEEE transactions on pattern analysis and machine intelligence*, 34(2):315–327, 2011.

[2] J. Angulo and D. Jeulin. Stochastic watershed segmentation. In *ISMM*. Springer, 2007.

[3] J. Angulo and J. Serra. Automatic analysis of dna microarray images using mathematical morphology. *Bioinformatics*, 19(5):553–562, 2003.

[4] J. Angulo, S. Velasco-Forero, and J. Chanussot. Multiscale stochastic watershed for unsupervised hyperspectral image segmentation. In *2009 IEEE International Geoscience and Remote Sensing Symposium*, volume 3, pages III–93. IEEE, 2009.

[5] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE PAMI*, 33(5):898–916, 2011.

[6] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE PAMI*, 33(5):898–916, May 2011.

[7] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Proceedings IEEE CVPR*, pages 328–335, 2014.

[8] R. Audigier and R. Lotufo. Seed-relative segmentation robustness of watershed and fuzzy connectedness approaches. In *XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2007)*, pages 61–70. IEEE, 2007.

[9] R. Audigier and R. Lotufo. Uniquely-determined thinning of the tie-zone watershed based on label frequency. *JMIV*, 27(2):157–173, 2007.

[10] R. Audigier and R. d. A. Lotufo. Watershed by image foresting transform, tie-zone, and theoretical relationships with other watershed definitions. *Mathematical*

Morphology and its Applications to Signal and Image Processing (ISMM), pages 277–288, 2007.

[11] J. A. Benediktsson, L. Bruzzone, J. Chanussot, M. Dalla Mura, P. Salembier, and S. Valero. Hierarchical analysis of remote sensing data: Morphological attribute profiles and binary partition trees. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 306–319. Springer, 2011.

[12] S. Beucher. Use of watersheds in contour detection. In *Proceedings of the International Workshop on Image Processing*. CCETT, 1979.

[13] S. Beucher. Watershed, hierarchical segmentation and waterfall algorithm. In *Mathematical morphology and its applications to image processing*, pages 69–76. Springer, 1994.

[14] S. Beucher and F. Meyer. The morphological approach to segmentation: the watershed transformation. *Optical Engineering-New York-Marcel Dekker Incorporated-*, 34:433–433, 1992.

[15] S. Beucher and F. Meyer. The morphological approach to segmentation: the watershed transformation. *Mathematical morphology in image processing*, 34:433–481, 1993.

[16] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive gmmrf model. In *European conference on computer vision*, pages 428–441. Springer, 2004.

[17] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 377–384. IEEE, 1999.

[18] E. J. Breen and R. Jones. Attribute openings, thinnings, and granulometries. *Computer Vision and Image Understanding*, 64(3):377–389, 1996.

[19] J. Chaussard, M. Couprie, and H. Talbot. Robust skeletonization using the discrete λ-medial axis. *Pattern Recognition Letters*, 32(9):1384–1394, 2011.

[20] C. Couprie, L. Grady, L. Najman, and H. Talbot. Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest. In *ICCV*, pages 731–738. IEEE, 2009.

[21] C. Couprie, L. Grady, L. Najman, and H. Talbot. Power watershed: A unifying graph-based optimization framework. *IEEE transactions on pattern analysis and machine intelligence*, 33(7):1384–1399, 2010.

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

[23] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts: Thinnings, shortest path forests, and topological watersheds. *IEEE PAMI*, 32(5):925–939, 2009.

[24] J. Cousty and L. Najman. Incremental algorithm for hierarchical minimum spanning forests and saliency of watershed cuts. In *ISMM*, pages 272–283. Springer, 2011.

[25] J. Cousty, L. Najman, Y. Kenmochi, and S. Guimarães. Hierarchical segmentations with graphs: quasi-flat zones, minimum spanning trees, and saliency maps. *JMIV*, 60(4):479–502, 2018.

[26] J. Cousty, L. Najman, Y. Kenmochi, and S. J. F. Guimarães. Hierarchical segmentations with graphs: quasi-flat zones, minimum spanning trees, and saliency maps. Technical report, 2016.

[27] J. Cousty, L. Najman, and B. Perret. Constructive links between some morphological hierarchies on edge-weighted graphs. In *ISMM*, pages 86–97. Springer, 2013.

[28] J. Cousty, L. Najman, and J. Serra. Raising in watershed lattices. In *2008 15th IEEE ICIP*, pages 2196–2199. IEEE, 2008.

[29] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1558–1570, 2014.

[30] L. Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, pages 128–140, 1741.

[31] A. X. Falcão, J. Stolfi, and R. de Alencar Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 26(1):19–29, 2004.

[32] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2012.

[33] A. Fehri, S. Velasco-Forero, and F. Meyer. Automatic selection of stochastic watershed hierarchies. In *EUSIPCO*, pages 1877–1881. IEEE, 2016.

[34] A. Fehri, S. Velasco-Forero, and F. Meyer. Characterizing images by the gromov-hausdorff distances between derived hierarchies. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1213–1217. IEEE, 2018.

[35] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.

[36] P. J. Figueroa, N. J. Leite, and R. M. Barros. A flexible software for tracking of markers used in human motion analysis. *Computer methods and programs in biomedicine*, 72(2):155–165, 2003.

[37] J. M. Gauch. Image segmentation and analysis via multiscale gradient watershed hierarchies. *IEEE transactions on image processing*, 8(1):69–79, 1999.

[38] D. M. Gavrila and V. Philomin. Real-time object detection for" smart" vehicles. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 87–93. IEEE, 1999.

[39] T. Géraud, E. Carlinet, S. Crozet, and L. Najman. A quasi-linear algorithm to compute the tree of shapes of nd images. In *International symposium on mathematical morphology and its applications to signal and image processing*, pages 98–110. Springer, 2013.

[40] P. Ghamisi, B. Höfle, and X. X. Zhu. Hyperspectral and lidar data fusion using extinction profiles and deep convolutional neural network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(6):3011–3024, 2016.

[41] P. Ghamisi, R. Souza, J. A. Benediktsson, X. X. Zhu, L. Rittner, and R. A. Lotufo. Extinction profiles for the classification of remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10):5631–5645, 2016.

[42] J. C. Gower and G. J. Ross. Minimum spanning trees and single linkage cluster analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 18(1):54–64, 1969.

[43] L. Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11):1768–1783, 2006.

[44] V. Grau, A. Mewes, M. Alcaniz, R. Kikinis, and S. K. Warfield. Improved watershed transform for medical image segmentation using prior information. *IEEE transactions on medical imaging*, 23(4):447–458, 2004.

[45] M. Grimaud. New measure of contrast: the dynamics. In *Image Algebra and Morphological Image Processing III*, volume 1769, pages 292–306. International Society for Optics and Photonics, 1992.

[46] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *2010 ieee computer society conference on computer vision and pattern recognition*, pages 2141–2148. IEEE, 2010.

[47] S. Guimarães, Y. Kenmochi, J. Cousty, Z. Patrocinio, and L. Najman. Hierarchizing graph-based image segmentation algorithms relying on region dissimilarity. *Mathematical Morphology-Theory and Applications*, 2(1):55–75, 2017.

[48] S. J. F. Guimarães, J. Cousty, Y. Kenmochi, and L. Najman. A hierarchical image segmentation algorithm based on an observation scale. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 116–125. Springer, 2012.

[49] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, number 1288, 2013.

[50] Q. Huang and B. Dom. Quantitative methods of evaluating image segmentation. In *Proceedings., International Conference on Image Processing*, volume 3, pages 53–56. IEEE, 1995.

[51] D. Jeulin. Morphological probabilistic hierarchies for texture segmentation. *Mathematical Morphology-Theory and Applications*, 1(1), 2016.

[52] B. R. Kiran and J. Serra. Fusion of ground truths and hierarchies of segmentations. *PRL*, 47:63–71, 2014.

[53] B. R. Kiran and J. Serra. Braids of partitions. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 217–228. Springer, 2015.

[54] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[55] R. Lotufo and W. Silva. Minimal set of markers for the watershed transform. In *Proceedings of ISMM*, volume 2002, pages 359–368, 2002.

[56] V. Machairas, M. Faessel, D. Cárdenas-Peña, T. Chabardes, T. Walter, and E. Decencière. Waterpixels. *IEEE Transactions on Image Processing*, 24(11):3707–3716, 2015.

[57] D. S. Maia, A. d. A. Araujo, J. Cousty, L. Najman, B. Perret, and H. Talbot. Evaluation of combinations of watershed hierarchies. In *ISMM*, pages 133–145. Springer, 2017.

[58] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. Van Gool. Convolutional oriented boundaries. In *ECCV*, pages 580–596. Springer, 2016.

[59] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. Van Gool. Convolutional oriented boundaries: From image segmentation to high-level tasks. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):819–833, 2017.

[60] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. Van Gool. Convolutional oriented boundaries: From image segmentation to high-level tasks. *PAMI*, 40(4):819–833, 2018.

[61] B. Marcotegui. Residual approach on a hierarchical segmentation. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4353–4357. IEEE, 2014.

[62] B. Marcotegui, F. Zanoguera, P. Correia, R. Rosa, F. Marqués, R. Mech, and M. Wollborn. A video object generation tool allowing friendly user interaction. In *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, volume 2, pages 391–395. IEEE, 1999.

[63] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[64] D. R. Martin. *An Empirical Approach to Grouping and Segmentation*. PhD thesis, University of California, 2003.

[65] E. Meinhardt. *Morphological and Statistical Techniques for the Analysis of 3D Images*. PhD thesis, Ph. D. thesis, Department of Information and Communication Technologies . . . , 2010.

[66] F. Meyer. Minimum spanning forests for morphological segmentation. In *Mathematical morphology and its applications to image processing*, pages 77–84. Springer, 1994.

[67] F. Meyer. The dynamics of minima and contours. In P. Maragos, R. Schafer, and M. Butt, editors, *ISMM*, pages 329–336. Kluwer, 1996.

[68] F. Meyer. Watersheds on weighted graphs. *Pattern Recognition Letters*, 47:72–79, 2014.

[69] F. Meyer and P. Maragos. Morphological scale-space representation with levelings. In *International Conference on Scale-Space Theories in Computer Vision*, Berlin, 1999. Springer.

[70] F. Meyer, C. Vachier, A. Oliveras, and P. Salembier. Morphological tools for segmentation: Connected filters and watersheds. In *Annales des télécommunications*, volume 52, pages 367–379. Springer, 1997.

[71] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[72] M. Nagao, T. Matsuyama, and Y. Ikeda. Region extraction and shape analysis in aerial photographs. *CGIP*, 10(3):195–223, 1979.

[73] L. Najman. On the equivalence between hierarchical segmentations and ultrametric watersheds. *JMIV*, 40(3):231–247, 2011.

[74] L. Najman, J. Cousty, and B. Perret. Playing with kruskal: algorithms for morphological trees in edge-weighted graphs. In *ISMM*, pages 135–146. Springer, 2013.

[75] L. Najman and M. Schmitt. Geodesic saliency of watershed contours and hierarchical segmentation. *IEEE PAMI*, 18(12):1163–1173, 1996.

[76] O. Okubadejo, E. Andò, L. Bonnaud, G. C. Viggiani, and M. Dalla Mura. Contact based hierarchical segmentation for granular materials. In *ISMM*, Saarbrücken, Germany, July 2019.

[77] G. K. Ouzounis and P. Soille. The alpha-tree algorithm. *Publications Office of the European Union*, 2012.

[78] T. Pavlidis. *Structural pattern recognition*, volume 2. Springer, 1977.

[79] B. Perret, J. Cousty, S. J. F. Guimaraes, and D. S. Maia. Evaluation of hierarchical watersheds. *IEEE TIP*, 27(4):1676–1688, 2018.

[80] B. Perret, J. Cousty, J. C. R. Ura, and S. J. F. Guimarães. Evaluation of morphological hierarchies for supervised segmentation. In *ISMM*, pages 39–50. Springer, 2015.

[81] J. Pont-Tuset, P. Arbeláez, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE PAMI*, 39(1):128–140, 2017.

[82] J. Pont-Tuset and F. Marques. Supervised assessment of segmentation hierarchies. *Computer Vision–ECCV 2012*, pages 814–827, 2012.

[83] J. Pont-Tuset and F. Marques. Upper-bound assessment of the spatial accuracy of hierarchical region-based image representations. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 865–868. IEEE, 2012.

[84] J. Pont-Tuset and F. Marques. Measures and meta-measures for the supervised evaluation of image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2131–2138, 2013.

[85] J. Pont-Tuset and F. Marques. Supervised evaluation of image segmentation and object proposal techniques. *IEEE PAMI*, 38(7):1465–1478, 2016.

[86] J. F. Randrianasoa, C. Kurtz, E. Desjardin, and N. Passat. Binary partition tree construction from multiple features for image segmentation. *Pattern Recognition*, 84:237–250, 2018.

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

[88] P. Salembier, A. Oliveras, and L. Garrido. Antiextensive connected operators for image and sequence processing. *TIP*, 7(4):555–570, 1998.

[89] P. Salembier and M. H. Wilkinson. Connected operators. *IEEE Signal Processing Magazine*, 26(6):136–157, 2009.

[90] J. Shi and J. Malik. Normalized cuts and image segmentation. *Departmental Papers (CIS)*, page 107, 2000.

[91] A. G. Silva and R. de Alencar Lotufo. New extinction values from efficient construction and analysis of extended attribute component tree. In *SIBGRAPI*, pages 204–211. IEEE, 2008.

[92] P. Soille. Constrained connectivity for hierarchical image partitioning and simplification. *IEEE transactions on pattern analysis and machine intelligence*, 30(7):1132–1145, 2008.

[93] R. Souza, L. Rittner, R. Machado, and R. Lotufo. A comparison between extinction filters and attribute filters. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 63–74. Springer, 2015.

[94] M. Stoer and F. Wagner. A simple min-cut algorithm. *Journal of the ACM (JACM)*, 44(4):585–591, 1997.

[95] C. Straehle, S. Peter, U. Köthe, and F. A. Hamprecht. K-smallest spanning tree segmentations. In *German Conference on Pattern Recognition*, pages 375–384. Springer, 2013.

[96] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Iccv*, volume 98, page 2, 1998.

[97] R. Trias-Sanz, G. Stamon, and J. Louchet. Using colour, texture, and hierarchial segmentation for high-resolution remote sensing. *ISPRS Journal of Photogrammetry and remote sensing*, 63(2):156–168, 2008.

[98] C. Vachier and F. Meyer. Extinction value: a new measurement of persistence. In *IEEE Workshop on nonlinear signal and image processing*, volume 1, pages 254–257, 1995.

[99] S. Valero, P. Salembier, and J. Chanussot. New hyperspectral data representation using binary partition tree. In *2010 IEEE International Geoscience and Remote Sensing Symposium*, pages 80–83. IEEE, 2010.

[100] V. Vilaplana, F. Marques, and P. Salembier. Binary partition trees for object detection. *IEEE Transactions on Image Processing*, 17(11):2201–2216, 2008.

[101] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):583–598, 1991.

[102] S. Wang and J. M. Siskind. Image segmentation with minimum mean cut. In *ICCV*, volume 1, pages 517–524. IEEE, 2001.

[103] J. H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[104] Y. Xu, E. Carlinet, T. Géraud, and L. Najman. Hierarchical segmentation using tree-based shape spaces. *IEEE PAMI*, 39(3):457–469, 2017.

[105] Y. Xu, T. Géraud, and L. Najman. Context-based energy estimator: Application to object segmentation on the tree of shapes. In *2012 19th IEEE International Conference on Image Processing*, pages 1577–1580. IEEE, 2012.

[106] Y. Xu, T. Géraud, and L. Najman. Morphological filtering in shape spaces: Applications using tree-based image representations. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 485–488. IEEE, 2012.

[107] C. T. Zahn. Graph theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Comput.*, 20(SLAC-PUB-0672-REV):68, 1970.

Appendix

8.1 Proofs of theorem and properties of Chapter 3

8.1.1 Proof of Lemma 19

(**Lemma 19).** *Let $\mathcal{H}$ be a hierarchy on V. The hierarchy $\mathcal{H}$ is a hierarchical watershed of (G, w) if and only if there is an altitude ordering $\prec$ for w such that the saliency map $\Phi(\mathcal{H})$ is one-side increasing for $\prec$.*

In order to prove Lemma 19, we will use an equivalent formulation provided in Property 12 of [27], revised in Section 2.6.1. We first remind a few notions presented in Section 2.6.1.

Let $\prec$ be an altitude ordering for w, let $\mathcal{B}_\prec$ be the binary partition hierarchy by $\prec$ and let $\mathcal{S} = (M_1, \ldots, M_n)$ be a sequence of minima of (G, w). Let X be a region of $\mathcal{B}_\prec$. Following the terminology of [27], the *extinction value of X for $\mathcal{S}$* is zero if there is no minimum M of w such that M is a subset of X and, otherwise, it is the highest index k such that M_k is a subset of X. Let ϵ be the map from the regions of $\mathcal{B}_\prec$ into $\mathbb{R}$ such that, for any region Y of $\mathcal{B}_\prec$, $\epsilon(Y)$ is the extinction map of Y for $\mathcal{S}$. We say that ϵ *is the extinction map (for $\prec$ and $\mathcal{S}$)* and that ϵ *is an extinction map* (for $\prec$).

Let $\prec$ be an altitude ordering for w, let $\mathcal{B}_\prec$ be the binary partition hierarchy by $\prec$ and let $\mathcal{S} = (M_1, \ldots, M_n)$ be a sequence of minima of w. Let u be a building edge for $\prec$ and let X be the region of $\mathcal{B}_\prec$ whose building edge is u. The *persistence value of u (for $\prec$ and $\mathcal{S}$)* is the minimum of the extinction values of the children of X. Let ρ be the map from the building edges for $\prec$ into $\mathbb{R}$ such that, for any building edge u of $\mathcal{B}_\prec$, $\rho(u)$ is the persistence value of u. We say that ρ *is the persistence map (for $\prec$ and $\mathcal{S}$)*. We denote by B_i the set of building edges of $\mathcal{B}_\prec$ whose persistence value is lower than or equal to i.

(**Definition 5). (hierarchy induced by an altitude ordering and a sequence of minima [27])** *Let $\prec$ be an altitude ordering for w, let $\mathcal{S} = (M_1, \ldots, M_n)$ be a sequence of minima of w and let ρ be the persistence map for $\prec$ and for $\mathcal{S}$. The sequence of* *partitions* $(CC(V, B_0), \ldots, CC(V, B_{n-1}))$ *is a hierarchy called* the hierarchy induced by $\prec$ and $\mathcal{S}$.

Recall that Lemma 19 states the equivalence between:

A A hierarchy $\mathcal{H}$ is a hierarchical watershed of (G, w).

B There exists an altitude ordering $\prec$ for w such that the saliency map $\Phi(\mathcal{H})$ is one-side increasing for $\prec$

In order to prove this theorem, we will use another equivalent property:

C There exists an altitude ordering $\prec$ for w and a sequence of minima $\mathcal{S}$ of w such that $\mathcal{H}$ is induced by $\prec$ and $\mathcal{S}$.

Property 12 of [27] established the equivalence between A and C. We will then show the equivalence between B and C. This proof is decomposed in Lemma 53 (C $\Rightarrow$ B) and Lemma 54 (B $\Rightarrow$ C).

We now introduce some new lemmas needed to prove Lemma 53.

Lemma 49. *Let $\prec$ be an altitude ordering for w and let ϵ be an extinction map for $\prec$. Let X and Y be two regions of $\mathcal{B}_\prec$. If $X \subseteq Y$, then $\epsilon(X) \leq \epsilon(Y)$.*

Proof. Since $\mathcal{B}_\prec$ is a hierarchy, we can affirm that, for any two regions Y and Z of $\mathcal{B}_\prec$, if $Y \subseteq Z$, then all minima of w included in Y are also included in Z and, therefore, $\epsilon(Y) \leq \epsilon(Z)$. □

Lemma 50. *Let $\prec$ be an altitude ordering for w and let ϵ be an extinction map for $\prec$. The range of ϵ is $\{0, \ldots, n\}$.*

Proof. To prove that the range of ϵ is $\{0, \ldots, n\}$, we will prove that $\{0, \ldots, n\} \subseteq range(\epsilon)$ and that $range(\epsilon) \subseteq \{0, \ldots, n\}$.

1. $\{0, \ldots, n\} \subseteq range(\epsilon)$. First, we prove that 0 is in $range(\epsilon)$. Let u be the lowest edge of $E_\prec$ for $\prec$. We can say that u is in a minimum of w. Moreover, the children of R_u are necessarily singletons. Hence, the extinction value of both children of u is zero. Now, we will prove for i in $\{1, \ldots, n\}$. By [74] (page 7), any minimum of w is a region of $\mathcal{B}$. Therefore, for any i in $\{1, \ldots, n\}$, there is a region of $\mathcal{B}_\prec$ whose extinction value is i.

2. $range(\epsilon) \subseteq \{0, \ldots, n\}$. For any region X of $\mathcal{B}_\prec$, if X contains at least one minimum of w, then its extinction value is in $\{1, \ldots, n\}$. Otherwise, the extinction value of X is zero.

□

Lemma 51. *Let $\prec$ be an altitude ordering on the edges of G for w, let $\mathcal{S} = (M_1, \ldots, M_n)$ be a sequence of minima of w and let ρ be the persistence map for $\prec$ and for $\mathcal{S}$. The range of ρ is $\{0, \ldots, n-1\}$.*

Proof. Let ϵ denote the extinction map for $\prec$ and $\mathcal{S}$. We will prove that (1) for any building edge u for $\prec$, $\rho(u)$ is in $\{0, \ldots, n-1\}$, and that, (2) for any i in $\{0, \ldots, n-1\}$, there is a building edge for $\prec$ whose persistence value is i.

1. $\{0, \ldots, n-1\} \subseteq range(\rho)$. First, we prove that 0 is in $range(\rho)$. By Lemma 50, there is a region X of $\mathcal{B}_\prec$ whose extinction value is zero. Therefore, the persistence value of the parent of X is equal to zero. Now, we will prove that any i in $\{1, \ldots, n-1\}$ is in $range(\rho)$. Let i be a value in $\{1, \ldots, n-1\}$. By [74] (page 7), the minimum M_i is a region of $\mathcal{B}_\prec$. Then, there is a region of $\mathcal{B}_\prec$ whose extinction value is i. Let X be the largest region of $\mathcal{B}_\prec$ whose extinction value is i. We can say that $X \neq V$ because M_n is included in V and, therefore, $\epsilon(V) = n$. Let Z be the parent of X. We can infer that the extinction value $\epsilon(Z)$ of Z is strictly greater than i. Therefore, there is a minimum M_j with $j > i$ included in the sibling of X. Hence, the extinction value of $sibling(X)$ is also strictly greater than i. Then, the persistence value of the building edge of Z, being the minimum of the extinction value of its children, is i.

2. $range(\rho) \subseteq \{0, \ldots, n-1\}$. Let u be an edge in $E(\mathcal{B}_\prec)$. By Lemma 50, and as the persistence value of u is equal to the extinction value of a child of R_u, we have that $\rho(u)$ is in $\{0, \ldots, n\}$. Moreover, the persistence value $\rho(u)$ of u is lower than n because, if the extinction value of one child X of R_u is n, then the minimum M_n is included in X and M_n is not included in $sibling(X)$, which implies that the extinction value of $sibling(X)$ is strictly lower than n. Therefore, since $\rho(u) = min\{\epsilon(X), \epsilon(sibling(X))\}$, the persistence value of u is strictly lower than n. Thus, we have that $range(\rho) \subseteq \{0, \ldots, n-1\}$.

□

Lemma 52. *Let $\prec$ be an altitude ordering for w, let $\mathcal{S} = (M_1, \ldots, M_n)$ be a sequence of minima of w and let ρ be the persistence map for $\prec$ and for $\mathcal{S}$. Let $\mathcal{H}$ be the hierarchy induced by $\prec$ and $\mathcal{S}$. For any building edge u for $\prec$, we have $\Phi(\mathcal{H})(u) = \rho(u)$.*

Proof. By definition, $\mathcal{H}$ is the sequence $(CC(V, B_0), \ldots, CC(V, B_{n-1}))$ such that, for any i in $\{0, \ldots, n-1\}$, B_i is the set of building edges for $\prec$ whose persistence values is lower than or equal to i. Let $u = \{x, y\}$ be a building edge for $\prec$ and let i be the persistence value of u. We can say that x and y are in the same region of $CC(V, B_i)$ but in distinct regions of $CC(V, B_{i-1})$ if $i \neq 0$. Therefore, since $CC(V, B_i)$ is the i-th partition of $\mathcal{H}$, by the definition of saliency maps, we have $\Phi(\mathcal{H})(u) = i$. □

Lemma 53. *Let $\prec$ be an altitude ordering for w, let $\mathcal{S}$ be a sequence of minima of w and let $\mathcal{H}$ be the hierarchy induced by $\prec$ and by $\mathcal{S}$. The saliency map $\Phi(\mathcal{H})$ of $\mathcal{H}$ is one-side increasing for $\prec$.*

Proof. Let $E_\prec$ be the set of building edges for $\prec$. In order to prove that $\Phi(\mathcal{H})$ is one-side increasing for $\prec$, by Definition 18, we need to prove that the following three statements hold true:

1. $\{\Phi(\mathcal{H})(e) \mid e \in E_\prec\} = \{0, \dots, n-1\}$;
2. for any edge u in $E_\prec$, $\Phi(\mathcal{H})(u) > 0$ if and only if u is a watershed-cut edge for $\prec$; and
3. for any edge u in $E_\prec$, there exists a child R of R_u such that $\Phi(\mathcal{H})(u) \geq \vee\{\Phi(\mathcal{H})(v)$ such that R_v is included in $R\}$, where $\vee\{\} = 0$.

In the sequel of this proof, let ρ and ϵ be respectively the persistence map and the extinction map for $\prec$ and $\mathcal{S}$.

1. By Lemma 82, we have $\{\Phi(\mathcal{H})(e) \mid e \in E_\prec\} = \{\rho(e) \mid e \in E_\prec\}$. Then, as Lemma 81 states that the range of ρ is $\{0, \dots, n-1\}$, we can conclude that $\{\Phi(\mathcal{H})(e) \mid e \in E_\prec\}$ is the set $\{0, \dots, n-1\}$.
2. Let u be a building edge for $\prec$. Given the following propositions:

   (a) u is a watershed-cut edge for $\prec$

   (b) $\Phi(\mathcal{H})(u) > 0$

   we will prove that (a) implies (b), and that not (b) implies not (a).

   If u is a watershed-cut edge for $\prec$, then both children of R_u contain at least one minimum of w. Therefore, the extinction value of both children of R_u is non-zero and, consequently, the persistence value $\rho(u)$ of u is non-zero. Moreover, by Lemma 82, in this case we have $\Phi(\mathcal{H})(e) = \rho(e)$ for any building edge e of $\mathcal{B}_\prec$. Thus, $\Phi(\mathcal{H})(u)$, being equal to $\rho(u)$, is non-zero.

   On the other hand, if u is not a watershed-cut edge for $\prec$, then there is a child X of R_u which does not contain any minimum of w. Therefore, the extinction value of X is equal to 0: $\epsilon(X) = 0$. Since, by definition $\rho(u) = min\{\epsilon(X), \epsilon(sibling(X))\}$ and the minimal extinction value is zero, we can say that $\rho(u) = 0$. Again, by Lemma 82, in this case we have $\Phi(\mathcal{H})(e) = \rho(e)$ for any building edge e of $\mathcal{B}_\prec$ and thus, $\Phi(\mathcal{H})(u)$, being equal to $\rho(u)$, is equal to 0.

3. Let u be a building edge of $\mathcal{B}_\prec$. The persistence value of u is the extinction value of a child X of R_u. Let X be a child of R_u such that $\rho(u)$, the persistence value of u, is equal to $\epsilon(X)$, the extinction value of X. By Lemma 49, for any region Y of $\mathcal{B}_\prec$ such that $Y \subseteq X$, we have $\epsilon(Y) \leq \epsilon(X)$ and, as $X \subseteq R_u$, $\epsilon(Y) \leq \epsilon(R_u)$. Let v be the building edge of a region $Z \subseteq X$. Then, we can say that the extinction value of both children of Z is less than or equal to the extinction value $\epsilon(X)$. Hence, $\rho(v) \leq \epsilon(X)$ and, then, $\rho(v) \leq \rho(u)$. By Lemma 82, we can conclude that $\Phi(\mathcal{H})(v) \leq \Phi(\mathcal{H})(u)$. Hence, $\Phi(\mathcal{H})(u) \geq \vee\{\Phi(\mathcal{H})(v)$ such that R_v is included in $X\}$.

□

Lemma 54. *Let $\prec$ be an altitude ordering of (G, w) and let $\mathcal{H}$ be a hierarchy on V such that $\Phi(\mathcal{H})$ is one-side increasing for $\prec$. Then there exists a sequence of minima $\mathcal{S}$ such that $\mathcal{H}$ is the hierarchy induced by $\prec$ and $\mathcal{S}$.*

In order to prove Lemma 54, we review Property 24, which is established in Chapter 4. The proof of Property 24 will be presented in Appendix 8.2 along with the proof of the other properties introduced in Chapter 4.

(**Property 24).** *Let $\mathcal{H}$ be a hierarchy. The hierarchy $\mathcal{H}$ is a hierarchical watershed of (G, w) if and only if there exists an altitude ordering $\prec$ for w and an extinction map ϵ for $\prec$ such that:*

1. *$(V, E_\prec)$ is a MST of $(G, \Phi(\mathcal{H}))$; and*

2. *for any edge u in $E_\prec$, we have that $\Phi(\mathcal{H})(u)$ is equal to* $\min\{\epsilon(R)$ *such that R is a child of $R_u\}$.*

The following lemma, established in [25], links MSTs and QFZ hierarchies.

Lemma 55 (Theorem 4 of [25]). *A subgraph G' of G is a MST of (G, w) if and only if:*

1. *the QFZ hierarchy of G' and G are the same; and*

2. *the graph G' is minimal for statement 1,* i.e., *for any subgraph G'' of G', if the quasi-flat zone hierarchy of G'' for w is the one of G for w, then we have $G'' = G'$.*

Lemma 56. *Let $\prec$ be an altitude ordering for (G, w) and let $\mathcal{H}$ be a hierarchy on V such that $\Phi(\mathcal{H})$ is one-side increasing for $\prec$. Let $E_\prec$ be the set of building edges for $\prec$. Then $(V, E_\prec)$ is a MST of $(G, \Phi(\mathcal{H}))$.*

Proof. Let α denote the sum of the weight of the edges in $E_\prec$ in the map $\Phi(\mathcal{H})$: $\alpha = \sum_{e \in E_\prec} \Phi(\mathcal{H})(e)$. As $\Phi(\mathcal{H})$ is one-side increasing for $\prec$, by the condition 1 of Definition 18, we can affirm that $\alpha = 0 + 1 + \cdots + n - 1$. In order to prove that $(V, E_\prec)$ is a MST of $(G, \Phi(\mathcal{H}))$, we will prove that, for any MST G' of $(G, \Phi(\mathcal{H}))$, the sum of the weight of the edges in G' is greater than or equal to α. Let G' be a MST of $(G, \Phi(\mathcal{H}))$. As G' is a MST of $(G, \Phi(\mathcal{H}))$, by the condition 1 of Lemma 83, we have that G and G' have the same quasi-flat zones hierarchy: $\mathcal{QFZ}(G, \Phi(\mathcal{H})) = \mathcal{QFZ}(G', \Phi(\mathcal{H}))$. As $\Phi(\mathcal{H})$ is the saliency map of $\mathcal{H}$, we have that $\mathcal{H} = \mathcal{QFZ}(G, \Phi(\mathcal{H}))$. Therefore, $\mathcal{H} = \mathcal{QFZ}(G', \Phi(\mathcal{H}))$. Let i be a value in $\{1, \ldots, n-1\}$. By the condition 1 of Definition 18, we can say that $\{1, \ldots, n-1\}$ is a subset of the range of $\Phi(\mathcal{H})$. Therefore, $\mathcal{H}$ is composed of at least n distinct partitions. Let $\mathcal{H}$ be the sequence $(\mathbf{P}_0, \ldots, \mathbf{P}_{n-1}, \ldots)$. Since the partitions $\mathbf{P}_i$ and $\mathbf{P}_{i-1}$ are distinct, then there exists a region in $\mathbf{P}_i$ which is not in $\mathbf{P}_{i-1}$. Therefore, there is a region X of $\mathbf{P}_i$ which is composed of several regions $\{R_1, R_2, \ldots\}$ of $\mathbf{P}_{i-1}$. Then, there are two adjacent vertices x and y such that x and y are in distinct regions in $\{R_1, R_2, \ldots\}$. Let x and y be two adjacent vertices such that x and y are in distinct regions in $\{R_1, R_2, \ldots\}$. Hence, the lowest j such that x and y belong to the same region of $\mathbf{P}_j$ is i. Thus, there exists an edge $u = \{x, y\}$ in $E_\prec$ such that $\Phi(\mathcal{H})(u) = i$. Hence, the sum of the weight of the edges of G' is at least $1 + \cdots + n - 1$, which is equal to α. Therefore, the graph $(V, E_\prec)$ is a MST of $(G, \Phi(\mathcal{H}))$. □

Proof of Lemma 54. As $\Phi(\mathcal{H})$ is one-side increasing for $\prec$, then, by Property 24, there is an extinction map ϵ such that, for any building edge u for $\prec$, $\Phi(\mathcal{H})(u) = min\{\epsilon(R)$ such that R is a child of $R_u\}$. Since ϵ is an extinction map, then there is a sequence $\mathcal{S} = (M_1, \ldots, M_n)$ of minima of w such that ϵ is the extinction map for $\prec$ and $\mathcal{S}$, and such that, for any region R of $\mathcal{B}_\prec$, $\epsilon(R) = \vee\{i \mid M_i \subseteq R\}$. Let G' denote the graph $(V, E_\prec)$. By Lemma 69, G' is a MST of $(G, \Phi(\mathcal{H}))$ and, consequently, by Lemma 83, $\mathcal{H} = \mathcal{QFZ}(G', \Phi(\mathcal{H}))$. Let ρ denote the persistence map for $\prec$ and for $\mathcal{S}$. Since $\Phi(\mathcal{H})(u) = min\{\epsilon(R)$ such that R is a child of $R_u\}$, we have that, for any building edge u, $\Phi(\mathcal{H})(u)$ is the persistence value $\rho(u)$ of u. Then, $\mathcal{QFZ}(G', \Phi(\mathcal{H})) = \mathcal{QFZ}(G', \rho)$. By definition, $\mathcal{QFZ}(G', \rho)$ is precisely the hierarchy induced by $\prec$ and by $\mathcal{S}$. □

8.1.2 Proof of Theorem 20

(**Theorem 20).** *Let $\mathcal{H}$ be a hierarchy on V and let $\prec$ be a lexicographic ordering for (w, f). The hierarchy $\mathcal{H}$ is a hierarchical watershed of (G, w) if and only if $\Phi(\mathcal{H})$ is one-side increasing for $\prec$.*

Let $\mathcal{H}$ be a hierarchy on V. By Theorem 19, $\mathcal{H}$ is a hierarchical watershed of (G, w) if and only if there is an altitude ordering for w such that the saliency map $\Phi(\mathcal{H})$ of $\mathcal{H}$ is one-side increasing for $\prec$. In order to prove Theorem 20, we will prove in the following lemma that, if the saliency map $\Phi(\mathcal{H})$ is one-side increasing for an altitude ordering for w, then $\Phi(\mathcal{H})$ is one-side increasing for any lexicographic ordering for $(w, \Phi(\mathcal{H}))$.

Given a map f from E into $\mathbb{R}$, we say that f *is a saliency map* if there is an hierarchy $\mathcal{H}$ on V such that f is the saliency map of $\mathcal{H}$.

Lemma 57. *Let f be a saliency map and let $\prec_f$ be a lexicographic ordering for (w, f). If there exists an altitude ordering $\prec$ for w such that f is one-side increasing for $\prec$, then f is one-side increasing for $\prec_f$.*

Let $\prec$ be an ordering on E and let $(u_1, \ldots, u_{|E|})$ be the sequence of edges in E such that, for any i in $\{1, \ldots, |E| - 1\}$, we have $u_i \prec u_{i+1}$. This sequence $(u_1, \ldots, u_{|E|})$ is called the *sequence (of edges) induced by* $\prec$. In order to prove Lemma 8.1.2, we first introduce the notion of *critical rank* and the notion of *switch* in the context of lexicographic orderings, and other auxiliary lemmas.

Definition 58 (critical rank)**.** *Let f be a saliency map and let $\prec$ be an altitude ordering for w. Let $(u_1, \ldots, u_{|E|})$ be the sequence induced by $\prec$. Let k be a value such that $u_k \prec u_{k+1}$ and such that $w(u_k) = w(u_{k+1})$ and $f(u_k) \geq f(u_{k+1})$. We say that k* is a critical rank for *f* and *$\prec$.*

Definition 59 (switch)**.** *Let f be a saliency map and let $\prec$ be an altitude ordering for w. Let $(u_1, \ldots, u_{|E|})$ be the sequence induced by $\prec$. Let k be a critical rank for f and $\prec$, and let $\prec_k$ be the ordering such that $(u_1, \ldots, u_{k+1}, u_k, \ldots, u_{|E|})$ is the sequence induced by $\prec_k$. We say that $\prec_k$ is a switch of $\prec$ for f (and k).*

Lemma 60. *Let f be a saliency map, let $\prec$ be an altitude ordering for w and let $\prec'$ be a switch of $\prec$ for f. Then $\prec'$ is an altitude ordering for w.*

Proof. Let $\prec'$ be the switch of $\prec$ for a critical rank k for f and $\prec$. Let $(u_1, \ldots, u_{|E|})$ be the sequence induced by $\prec$. Then $(u_1, \ldots, u_{k+1}, u_k, \ldots, u_{|E|})$ is the sequence induced by $\prec'$. We may affirm that, for any edge v different from u_{k+1}, if $v \prec u_k$ (resp. $u_k \prec v$)

then $v \prec' u_k$ (resp. $u_k \prec' v$). Similarly, for any edge v different from u_k, if $v \prec u_{k+1}$ (resp. $u_{k+1} \prec v$) then $v \prec' u_{k+1}$ ($u_{k+1} \prec' v$). Finally, for any two edges u and v such that $\{u, v\} \cap \{u_k, u_{k+1}\} = \emptyset$, if $u \prec v$ (resp. $v \prec u$), then $u \prec' v$ (resp. $v \prec' u$). Hence, for any two edges u and v such that $w(u) < w(v)$, by the definition of critical rank, we may say that $\{u, v\} \neq \{u_k, u_{k+1}\}$ and, consequently, as $u \prec v$, then $u \prec' v$. Hence, $\prec'$ is an altitude ordering for w. □

Lemma 61. *Let $\prec$ be an altitude ordering for w and let f be a saliency map. Let $\prec'$ be a lexicographic ordering for (w, f). There exists a sequence $(\prec_0, \prec_1, \ldots, \prec_\ell)$ of altitude orderings for w such that $\prec_0$ is equal to $\prec$, $\prec_\ell$ is equal to $\prec'$ and, for any i in $\{1, \ldots, \ell\}$, $\prec_i$ is a switch of $\prec_{i-1}$.*

Proof. Let $(u_1, \ldots, u_{|E|})$ be the sequence induced by $\prec$ and let $(u'_1, \ldots, u'_{|E|})$ be the sequence induced by $\prec'$. Let k be the smallest value such that $u_k \neq u'_k$. In this case, there is an $i > k$ such that $u'_k = u_i$. As $\prec'$ is a lexicographic ordering for (w, f), for any edge u_j such that $k < j \leq i$, we have $f(u_j) \geq f(u_{j-1})$. Hence, there is a sequence S of switches for critical ranks ranging from $i-1$ to k such that, in the last ordering $\prec^*$ of the sequence S, the edge with rank k for the ordering $\prec^*$ is precisely the edge u'_k. Let $(u^*_1, \ldots, u^*_{|E|})$ be the sequence induced by $\prec^*$. We conclude that, for any $i \leq k$, we have $u^*_k = u'_k$. Hence, the smallest value m such that $u^*_m \neq u'_m$ is strictly greater than k. By performing this procedure iteratively (like the bubble sort algorithm), the resulting ordering converge to $\prec'$. □

Lemma 62. *Let $\prec$ be an altitude ordering for w and let f be a saliency map such that f is one-side increasing for $\prec$. Let v_1 and v_2 be two edges of E. If $f(v_1)$ is equal to $f(v_2)$, then neither v_1 nor v_2 is a watershed-cut edge for $\prec$.*

Proof. Since f is one-side increasing for $\prec$, by Definition 18, we have $\{f(u) \mid u \in E_\prec\} = \{0, \ldots, n-1\}$ and we have that, for any edge u in $E_\prec$, $f(u)$ is greater than 0 if and only if u is a watershed-cut edge for $\prec$. Since w has n minima, there are $n-1$ watershed-cut edges for $\prec$. Hence, the watershed-cut edges for $\prec$ have pairwise distinct edge weights ranging from 1 to $n-1$. Therefore, neither v_1 nor v_2 is a watershed-cut edge for $\prec$. □

Let $\prec$ be an altitude ordering for w and let f be a saliency map such that f is one-side increasing for $\prec$. By Lemma 60, every switch of $\prec$ is an altitude ordering for w. By Lemma 61, any lexicographic ordering for (w, f) can be obtained by a sequence of switches starting from $\prec$. Hence, to prove Lemma , we can simply prove that f is one-side increasing for any switch of $\prec$. Let $(u_1, \ldots, u_{|E|})$ be the sequence induced by $\prec$.

Then $(u_1, \ldots, u_{k+1}, u_k, \ldots, u_{|E|})$ is the sequence induced by $\prec'$. In order to prove that f is one-side increasing for the switch $\prec'$ for k, we should consider the following cases:

1. Neither u_k nor u_{k+1} is a building edge for $\prec$;
2. Both u_k and u_{k+1} are building edges for $\prec$ and $R_{u_k} \cap R_{u_{k+1}} = \emptyset$;
3. Both u_k and u_{k+1} are building edges for $\prec$ and $R_{u_k} \subset R_{u_{k+1}}$;
4. Only u_{k+1} is a building edge for $\prec$; and
5. Only u_k is a building edge for $\prec$.

The following lemmas 64, 65, 66, 67 and 68 prove that, for each of those five cases, the saliency map f is one-side increasing for the switch $\prec'$ for k. Before considering those five cases, we first present the following auxiliary lemma.

Lemma 63. *Let $\prec$ be an altitude ordering for w and let f be a saliency map such that f is one-side increasing for $\prec$. Let $\prec'$ be an altitude ordering for w such that the set of building edges for $\prec'$ is equal to the set of building edges for $\prec$ and such that the set of regions of $\mathcal{B}_\prec$ is equal to the set of regions of $\mathcal{B}_{\prec'}$. Then f is one-side increasing for $\prec'$.*

Proof. In the definition of one-side increasing maps (Definition 18), the three conditions for f to be one-side increasing for $\prec$ take into consideration only the weight of the building edges for $\prec$ and parenthood relationship between the regions of $\prec$. Hence, as the set of building edges for $\prec'$ is the same set of building edges for $\prec$ and as they have the same set of regions, we can conclude that the three conditions of Definition 18 for f to the one-side increasing for $\prec'$ are satisfied. □

Lemma 64. *Let $\prec$ be an altitude ordering for w and let f be a saliency map such that f is one-side increasing for $\prec$. Let $(u_1, \ldots, u_{|E|})$ be the sequence induced by $\prec$. Let k be a critical rank for f and $\prec$ such that neither u_k nor u_{k+1} is a building edge for $\prec$. Then f is one-side increasing for the switch $\prec'$ for k.*

Proof. Let $(\mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_{|E|})$ be the sequence of partitions (of V) such that, for any i in $\{1, \ldots, |E|\}$, the partition $\mathbf{B}_i$ is the i-partition by the ordering $\prec$ (as defined in Section 2.5.2). Let $(\mathbf{B}'_0, \mathbf{B}'_1, \ldots, \mathbf{B}'_{|E|})$ be the sequence of partitions such that, for any i in $\{1, \ldots, |E|\}$, the partition $\mathbf{B}'_i$ is the i-partition by the ordering $\prec'$.

By the definition of binary partition hierarchy and, as neither u_k nor u_{k+1} is a building edge for $\prec$, we may say that:

I the partition $\mathbf{B}_k$ is equal to the partition $\mathbf{B}_{k-1}$, and

II the partition $\mathbf{B}_{k+1}$ is equal to the partition $\mathbf{B}_k$,

III which implies that $\mathbf{B}_{k-1} = \mathbf{B}_k = \mathbf{B}_{k+1}$.

Let $u_k = \{s, r\}$ and $u_{k+1} = \{x, y\}$. By the definition of switch, the sequence $(u_1, \ldots, u_{k+1}, u_k, \ldots, u_{|E})$ is the sequence induced by $\prec'$. We may infer that, for any $i < k$, the i-partition by the ordering $\prec'$ is equal to the i-partition by the ordering $\prec$. Hence, as u_{k+1} is the edge of rank k for $\prec'$ and since $\mathbf{B}'_{k-1} = \mathbf{B}_{k-1}$, the k-partition for the ordering $\prec'$ is the partition $\mathbf{B}'_k = \{\mathbf{B}^y_{k-1} \cup \mathbf{B}^x_{k-1}\} \cup (\mathbf{B}_{k-1} \setminus \{\mathbf{B}^x_{k-1}, \mathbf{B}^y_{k-1}\})$. By the statement I, $\mathbf{B}_{k-1} = \mathbf{B}_k$, which implies that $\mathbf{B}'_k = \{\mathbf{B}^y_k \cup \mathbf{B}^x_k\} \cup (\mathbf{B}_k \setminus \{\mathbf{B}^x_k, \mathbf{B}^y_k\})$. Therefore, we have that:

IV $\mathbf{B}'_k$ is equal to the partition $\mathbf{B}_{k+1}$

As $\mathbf{B}_{k+1} = \mathbf{B}_k = \mathbf{B}_{k-1}$ by statement III, we have that

V $\mathbf{B}'_k = \mathbf{B}_{k+1} = \mathbf{B}_{k-1} = \mathbf{B}'_{k-1}$

By statement V, as $\mathbf{B}'_k = \mathbf{B}'_{k-1}$, we conclude that u_{k+1} is not a building edge for $\prec'$.

Now, as u_k is the edge of rank $k+1$ for $\prec'$, the $k+1$-partition for the ordering $\prec'$ is the partition $\mathbf{B}'_{k+1} = \{\mathbf{B}'^s_k \cup \mathbf{B}'^r_k\} \cup (\mathbf{B}'_k \setminus \{\mathbf{B}'^s_k, \mathbf{B}'^r_k\})$. By statement V, we have $\mathbf{B}'_k = \mathbf{B}'_{k-1}$. Since $\mathbf{B}'_{k-1} = \mathbf{B}_{k-1}$, then, by statement III, we have that $\mathbf{B}'_k = \mathbf{B}_{k-1}$. Therefore, we conclude that:

VI $\mathbf{B}'_{k+1} = \{\mathbf{B}^s_{k-1} \cup \mathbf{B}^r_{k-1}\} \cup (\mathbf{B}_{k-1} \setminus \{\mathbf{B}^s_{k-1}, \mathbf{B}^r_{k-1}\})$

By the definition of $\mathbf{B}'_{k+1}$ in the statement VI, we have:

VII $\mathbf{B}'_{k+1} = \mathbf{B}_k$

By statement IV, $\mathbf{B}'_k = \mathbf{B}_{k+1}$ and, by statement III, $\mathbf{B}_k = \mathbf{B}_{k+1}$. Hence, $\mathbf{B}_k = \mathbf{B}'_k$. Thus, by the statement VII, we conclude that $\mathbf{B}'_{k+1} = \mathbf{B}'_k$. Therefore, u_k is not a building edge for $\prec'$.

Since the sequences induced by the orderings $\prec$ and $\prec'$ are equal for any $i > k+1$, and since $\mathbf{B}'_{k+1} = \mathbf{B}'_k = \mathbf{B}_k = \mathbf{B}_{k+1}$, we may affirm that, $\mathbf{B}_i = \mathbf{B}'_i$ for any $i > k+1$. Therefore, the set of building edges for $\prec$ is equal to the set of building edges for $\prec'$, and the set of partitions and regions of $\mathcal{B}_\prec$ is equal to the set of partitions and regions of $\mathcal{B}_{\prec'}$. By Lemma 63, f is one-side increasing for $\prec'$. □

Lemma 65. *Let $\prec$ be an altitude ordering for w and let f be a saliency map such that f is one-side increasing for $\prec$. Let $(u_1, \ldots, u_{|E|})$ be the sequence induced by $\prec$. Let k be a critical rank for f and $\prec$ such that both u_k and u_{k+1} are building edges for $\prec$ and such that $R_{u_k} \cap R_{u_{k+1}} = \emptyset$. Then f is one-side increasing for the switch $\prec'$ for k.*

Proof. Let $(\mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_{|E|})$ be the sequence of partitions (of V) such that, for any i in $\{1, \ldots, |E|\}$, the partition $\mathbf{B}_i$ is the i-partition by the ordering $\prec$. Let $(\mathbf{B}'_0, \mathbf{B}'_1, \ldots, \mathbf{B}'_{|E|})$ be the sequence of partitions such that, for any i in $\{1, \ldots, |E|\}$, the partition $\mathbf{B}'_i$ is the i-partition by the ordering $\prec'$. By the definition of switch, the sequence $(u_1, \ldots, u_{k+1}, u_k, \ldots, u_{|E})$ is the sequence induced by $\prec'$. As the sequences induced by $\prec$ and by $\prec'$ are equal for any edge with rank $i < k$, we may affirm that:

I $\mathbf{B}_i = \mathbf{B}'_i$ for any $i < k$

Let $u_k = \{s, r\}$ and $u_{k+1} = \{x, y\}$. As u_k and u_{k+1} are building edges for $\prec$, we have that:

II $\mathbf{B}_k \neq \mathbf{B}_{k-1}$, and

III $\mathbf{B}_{k+1} \neq \mathbf{B}_k$

As u_{k+1} is the edge of rank k for $\prec'$, we have that the k-partition for the ordering $\prec'$ is $\mathbf{B}'_k = \{{\mathbf{B}'_{k-1}}^x \cup {\mathbf{B}'_{k-1}}^y\} \cup (\mathbf{B}'_{k-1} \setminus \{{\mathbf{B}'_{k-1}}^x, {\mathbf{B}'_{k-1}}^y\})$. By the statement I, $\mathbf{B}'_{k-1}$ and $\mathbf{B}_{k-1}$ are equal. Then $\mathbf{B}'_k = \{\mathbf{B}^x_{k-1} \cup \mathbf{B}^y_{k-1}\} \cup (\mathbf{B}_{k-1} \setminus \{\mathbf{B}^x_{k-1}, \mathbf{B}^y_{k-1}\})$.

By definition, we have:

IV $\mathbf{B}_k = \{\mathbf{B}^s_{k-1} \cup \mathbf{B}^r_{k-1}\} \cup (\mathbf{B}_{k-1} \setminus \{\mathbf{B}^s_{k-1}, \mathbf{B}^r_{k-1}\})$, and

V $\mathbf{B}_{k+1} = \{\mathbf{B}^x_k \cup \mathbf{B}^y_k\} \cup (\mathbf{B}_k \setminus \{\mathbf{B}^x_k, \mathbf{B}^y_k\})$

By our hypothesis, $R_{u_k} \cap R_{u_{k+1}} = \emptyset$, which means that the regions R_{u_k} and $R_{u_{k+1}}$ of $\mathcal{B}_\prec$ (whose building edges are respectively u_k and u_{k+1}) have no intersection. As u_k is a building edge for $\prec$, we have $R_{u_k} = \{\mathbf{B}^s_{k-1} \cup \mathbf{B}^r_{k-1}\}$. Similarly, as u_{k+1} is a building edge for $\prec$, we have $R_{u_{k+1}} = \{\mathbf{B}^x_k \cup \mathbf{B}^y_k\}$. Since $R_{u_k} \cap R_{u_{k+1}} = \emptyset$, we have that:

VI neither x nor y is in the region $\mathbf{B}^s_{k-1}$ (resp. $\mathbf{B}^r_{k-1}$), and

VII neither s nor r is in the region $\mathbf{B}^x_k$ (resp. $\mathbf{B}^y_k$)

By VI and VII, we can conclude that $\mathbf{B}^s_{k-1}$, $\mathbf{B}^r_{k-1}$, $\mathbf{B}^x_k$ and $\mathbf{B}^y_k$ are all distinct regions of the partition $\mathbf{B}_{k-1}$. Hence, we have:

VIII $\mathbf{B}_k^x = \mathbf{B}_{k-1}^x$, and

IX $\mathbf{B}_k^y = \mathbf{B}_{k-1}^y$

By definition, as u_{k+1} is the edge of rank k for $\prec'$, we have:

X $\mathbf{B}'_k = \{\mathbf{B}'_{k-1}{}^x \cup \mathbf{B}'_{k-1}{}^y\} \cup (\mathbf{B}'_{k-1} \setminus \{\mathbf{B}'_{k-1}{}^x, \mathbf{B}'_{k-1}{}^y\})$

By I and X, we conclude that:

XI $\mathbf{B}'_k = \{\mathbf{B}_{k-1}^x \cup \mathbf{B}_{k-1}^y\} \cup (\mathbf{B}_{k-1} \setminus \{\mathbf{B}_{k-1}^x, \mathbf{B}_{k-1}^y\})$

By VIII, IX and XI, we conclude:

XII $\mathbf{B}'_k = \{\mathbf{B}_k^x \cup \mathbf{B}_k^y\} \cup (\mathbf{B}_k \setminus \{\mathbf{B}_k^x, \mathbf{B}_k^y\})$

As $\mathbf{B}_k^x$ and $\mathbf{B}_k^y$ are distinct regions, we may say that $\mathbf{B}'_k$ is different from $\mathbf{B}'_{k-1}$. Hence, u_{k+1} is a building edge for $\prec'$.

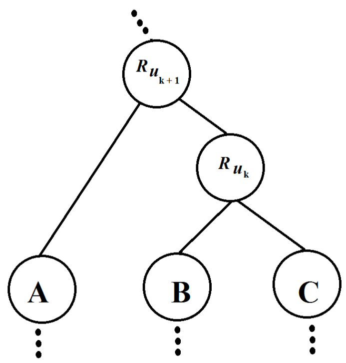
Now, as u_k is the edge of rank $k+1$ for $\prec'$, we have that the $(k+1)$-partition for the ordering $\prec'$ is $\mathbf{B}'_{k+1} = \{\mathbf{B}'_k{}^s \cup \mathbf{B}'_k{}^r\} \cup (\mathbf{B}'_k \setminus \{\mathbf{B}'_k{}^s, \mathbf{B}'_k{}^r\})$. By statement VII, we have that neither s nor r are in the regions $\mathbf{B}_k^x$ and $\mathbf{B}_k^y$. Hence, by the statement XII, s and r belong to distinct regions of $\mathbf{B}'_k$. Therefore, $\mathbf{B}'_k{}^s \neq \mathbf{B}'_k{}^r$. Consequently, $\mathbf{B}'_{k+1}$ is different from $\mathbf{B}'_k$. Hence, u_k is a building edge for $\prec'$.

Moreover, we conclude that $\mathbf{B}'_{k+1} = \mathbf{B}_{k+1}$ because both partitions result from the union of the four distinct regions of $\mathbf{B}_{k-1}$ containing s, r, x and y. Hence, for any $i > k+1$, as the sequences induced by $\prec$ and $\prec'$ are equal, we can conclude that any partition $\mathbf{B}_i$ is equal to the partition $\mathbf{B}'_i$ for any $i > k+1$. Therefore, by Lemma 63, f is one-side increasing for $\prec'$.

□

Lemma 66. *Let $\prec$ be an altitude ordering for w and let f be a saliency map such that f is one-side increasing for $\prec$. Let $(u_1, \ldots, u_{|E|})$ be the sequence induced by $\prec$. Let k be a critical rank for f and $\prec$ such that both u_k and u_{k+1} are building edges for $\prec$ and such that $R_{u_k} \subset R_{u_{k+1}}$. Then f is one-side increasing for the switch $\prec'$ for k.*

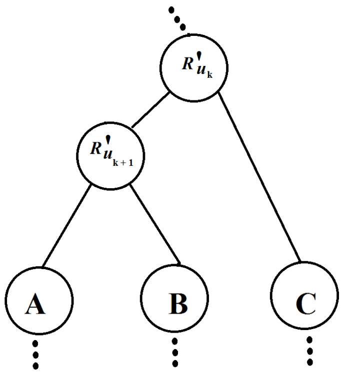
Proof. By our hypothesis, the region R_{u_k} of $\mathcal{B}_\prec$ is a subset of the region $R_{u_{k+1}}$ of $\mathcal{B}_\prec$. Let A be the region of $\mathcal{B}_\prec$ such that $R_{u_{k+1}} = R_{u_k} \cup A$. Let B and C be the children of R_{u_k}. This situation is illustrated in the following figure.



$\mathcal{B}_{\prec}$

Let $u_k = \{s, r\}$ and $u_{k+1} = \{x, y\}$. As u_{k+1} is a building edge for $\prec$, we conclude that x are y belong to two distinct regions in $\{A, B, C\}$. Without loss of generality, let us assume that x belongs to A and that y belongs to B. Let $\mathbf{B}_{k-1}$ be the $(k-1)$-partition for $\prec$. We can say that the regions A, B and C belong to $\mathbf{B}_{k-1}$. Moreover, we know that $\mathbf{B}_{k-1}$ is equal to the $(k-1)$-partition for $\prec'$ because, for any $i < k$, the edge of rank i for $\prec$ is also the edge of rank i for $\prec'$. Since u_{k+1} is the edge of rank k for $\prec'$, we can conclude that the k-partition $\mathbf{B}'_k$ for $\prec'$ is the partition $\{A \cup B\} \cup (\mathbf{B}_{k-1} \setminus \{A, B\})$. As the region $\{A \cup B\}$ is not in the partition $\mathbf{B}'_{k-1}$, we can conclude that $\mathbf{B}'_k$ is different from $\mathbf{B}'_{k-1}$. Hence, u_{k+1} is the building edge of the region $R'_{u_{k+1}} = \{A \cup B\}$ of $\mathcal{B}_{\prec'}$.

Now, without loss of generality, let us assume that s belongs to B and that r belongs to C. By our hypothesis, u_k is the edge of rank $k+1$ for $\prec'$. In the partition $\mathbf{B}'_k$, we know that s and r belong to distinct regions because s is in $\{A \cup B\}$ and r is in C. Hence, the region $\{A \cup B \cup C\}$ is a region of $\mathbf{B}'_{k+1}$ and we have $\mathbf{B}'_{k+1} \neq \mathbf{B}'_k$. Therefore, u_k is a building edge for $\prec'$. This situation is illustrated in the following figure.



$\mathcal{B}_{\prec'}$

We can infer that the $(k+1)$-partition for $\prec'$ is equal to the $(k+1)$-partition for $\prec$. Since the edge of rank i for $\prec$ is also the edge of rank i for $\prec'$, we can conclude that the set of building edges for $\prec$ is equal to the set of building edges for $\prec'$.

Now, we will prove that f is one-side increasing for $\prec'$. To that end, we will demonstrated that the three conditions of the definition of one-side increasing maps (Definition 18) hold true for f.

1. We first prove that the condition 1 of Definition 18 holds true for f. Since the set $E_{\prec}$ of building edges for $\prec$ is equal to the set $E_{\prec'}$ of building edges for $\prec'$, we can conclude that $\{f(u) \mid u \in E_{\prec'}\}$ is equal to $\{f(u) \mid u \in E_{\prec}\} = \{0, \ldots, n-1\}$. Thus, the first condition for f to be one-side increasing for $\prec'$ holds true.

2. We now prove that the condition 2 of Definition 18 holds true for f.

   In order to prove this condition, we consider four cases: (2.1) both u_k and u_{k+1} are watershed-cut edges for $\prec$; (2.2) neither u_k nor u_{k+1} is a watershed-cut edge for $\prec$; (2.3) only u_k is a watershed-cut for $\prec$; and (2.4) only u_{k+1} is a watershed-cut for $\prec$.

   (2.1) If both u_k and u_{k+1} are watershed-cut edges for $\prec$, then there is at least one minimum of w included in each of the regions A, B and C. Since A and B are the children of $R'_{u_{k+1}}$, we may say that u_{k+1} is a watershed-cut edge for $\prec'$. Since $\{A \cup B\}$ and C are the children of R'_{u_k} and since there is at least one minimum included in each of the children of R'_{u_k}, we may say that u_k is a watershed-cut edge for $\prec'$. Hence, both u_k and u_{k+1} are watershed-cut edges for $\prec'$.

(2.2) If neither u_k nor u_{k+1} is a watershed-cut edge for $\prec$, then there are at least two regions among A, B and C that do not include any minimum of w. Hence, there is at least one child of each of the regions R'_{u_k} and $R'_{u_{k+1}}$ that do not include any minimum of w. Hence, neither u_k nor u_{k+1} is a watershed-cut edge for $\prec'$.

(2.3) If u_k is a watershed-cut edge for $\prec$ and if u_{k+1} is not watershed-cut edge for $\prec$, then there is at least one minimum included in each of the regions B and C and there is no minimum included in A. Hence, as A is a child of the region $R'_{u_{k+1}}$ of $\mathcal{B}_{\prec'}$ and as there is no minimum of w included in A, u_{k+1} is not a watershed-cut edge for $\prec'$. Since there is at least one minimum included in each of the regions B and C, and since B and C are included in distinct children of the region R'_{u_k}, we can conclude that u_k is a watershed-cut edge for $\prec'$.

(2.4) If u_{k+1} is a watershed-cut edge for $\prec$ and if u_k is not watershed-cut edge for $\prec$. As k is a critical rank for f and $\prec$, we have that $f(u_k) \geq f(u_{k+1})$. However, by the definition of one-side increasing maps (Definition 18), we have $f(u_{k+1}) > 0$ and $f(u_k) = 0$, which contradicts our hypothesis. Therefore, the case where u_{k+1} is a watershed-cut edge for $\prec$ and if u_k is not watershed-cut edge for $\prec$ does not happen.

Therefore, we can conclude that the set of watershed-cut edges for $\prec$ is equal to the set of watershed-cut edges for $\prec'$. Then, the second condition for f to be one-side increasing for $\prec'$ holds true.

3. We finally prove that the condition 3 of Definition 18 holds true for f. As k is a critical rank for f and $\prec$, we have that $f(u_k) \geq f(u_{k+1})$. We will consider two cases: (3.1) $f(u_k) = f(u_{k+1})$; and (3.2) $f(u_k) > f(u_{k+1})$.

(3.1) If $f(u_k) = f(u_{k+1})$, by Lemma 62, neither u_k nor u_{k+1} is a watershed-cut edge for $\prec$. Since neither u_k nor u_{k+1} is a watershed-cut edge for $\prec$, as proven in the case (2.2), neither u_k nor u_{k+1} is a watershed-cut edge for $\prec'$. Hence, there is at least one child of the region R'_{u_k} (resp. $R'_{u_{k+1}}$) that does not include any minimum of w. Let Z be the child of R'_{u_k} (resp. $R'_{u_{k+1}}$) that does not include any minimum of w. We can infer that there is no watershed-cut edge v for $\prec'$ such that $R_v \subseteq Z$. Then, for any edge v such that $R_v \subseteq Z$, we have $f(v) = 0$. Since $f(u_k) = 0$ (resp. $f(u_{k+1}) = 0$), we can affirm that there is a child Z of R'_{u_k} (resp. $R'_{u_{k+1}}$) such that $f(u_k) \geq \{f(v) \mid R_v \subseteq Z\}$ (resp. $f(u_{k+1}) \geq \{f(v) \mid R_v \subseteq Z\}$).

(3.2) Let us assume that $f(u_k) > f(u_{k+1})$. Since f is one-side increasing for $\prec$, by Definition 18 (statement 3), we conclude that, for any edge v such that v is the building edge of a region included in A, we have $f(u_{k+1}) \geq f(v)$. In the hierarchy $\mathcal{B}_{\prec'}$, the region $R'_{u_{k+1}}$ is the parent of A, so the statement 3 of Definition 18 holds true for $R'_{u_{k+1}}$.

We will now prove that the statement 3 of Definition 18 holds true for R'_{u_k}. By Definition 18, we know that there is a child Z of R_{u_k} such that for any edge v such that v is the building edge of a region included in Z, we have $f(u_k) \geq f(v)$. Let us first assume that $Z = C$. Since C is also a child of the region R'_{u_k} of $\mathcal{B}_{\prec'}$, the statement 3 of Definition 18 holds true for R'_{u_k}. Now, let us assume that $Z = B$. We will prove that, for the building edge v of any region included in $\{A \cup B \cup R_{u_{k+1}}\}$, we have $f(u_k) \geq f(v)$. By our assumption $f(u_k) > f(u_{k+1})$. Moreover, for any edge v such that v is the building edge of a region included in A, we have $f(u_{k+1}) \geq f(v)$. Therefore, for the building edge v of any region included in $\{A \cup B \cup R_{u_{k+1}}\}$, we have $f(u_k) \geq f(v)$.

□

Lemma 67. *Let $\prec$ be an altitude ordering for w and let f be a saliency map such that f is one-side increasing for $\prec$. Let $(u_1, \ldots, u_{|E|})$ be the sequence induced by $\prec$. Let k be a critical rank for f and $\prec$ such that u_{k+1} is a building edge for $\prec$ and such that u_k is not a building edge for $\prec$. Then f is one-side increasing for the switch $\prec'$ for k.*

Proof. Let $(\mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_{|E|})$ be the sequence of partitions (of V) such that, for any i in $\{1, \ldots, |E|\}$, the partition $\mathbf{B}_i$ is the i-partition by the ordering $\prec$ (as defined in Section 3.1). Let $(\mathbf{B}'_0, \mathbf{B}'_1, \ldots, \mathbf{B}'_{|E|})$ be the sequence of partitions such that, for any i in $\{1, \ldots, |E|\}$, the partition $\mathbf{B}'_i$ is the i-partition by the ordering $\prec'$. As the sequences induced by $\prec$ and by $\prec'$ are equal for any edge with rank $i < k$, we may affirm that:

I. $\mathbf{B}_i = \mathbf{B}'_i$ for any $i < k$

By the definition of binary partition hierarchy and since u_k is not a building edge for $\prec$, we may say that:

II. the partition $\mathbf{B}_k$ is equal to the partition $\mathbf{B}_{k-1}$.

Let $u_k = \{s, r\}$ and $u_{k+1} = \{x, y\}$. Since $\mathbf{B}_k = \mathbf{B}_{k-1}$ and since $\mathbf{B}_k = \{\mathbf{B}^s_{k-1} \cup \mathbf{B}^r_{k-1}\} \cup (\mathbf{B}_{k-1} \setminus \{\mathbf{B}^s_{k-1}, \mathbf{B}^r_{k-1}\})$, we conclude that the regions $\mathbf{B}^s_{k-1}$ and $\mathbf{B}^r_{k-1}$ of the partition $\mathbf{B}_{k-1}$ are equal: $\mathbf{B}^s_{k-1} = \mathbf{B}^r_{k-1}$. By the statement I, we may say that the regions ${\mathbf{B}'_{k-1}}^s$ and ${\mathbf{B}'_{k-1}}^r$ of the partition $\mathbf{B}'_{k-1}$ are equal as well. Hence:

III. the partition $\mathbf{B}'_k$ is equal to the partition and $\mathbf{B}'_{k-1}$

Therefore, u_k is not a building edge for $\prec'$.

Since u_{k+1} is a building edge for $\prec$, we have that:

IV. the partition $\mathbf{B}_{k+1}$ is different from the partition $\mathbf{B}_k$.

By the statement IV, we conclude that the regions $\mathbf{B}^x_k$ and $\mathbf{B}^y_k$ of the partition $\mathbf{B}_k$ are distinct. By the statement III, we have that $\mathbf{B}'_k = \mathbf{B}'_{k-1}$. Then, by statement I, we have $\mathbf{B}'_k = \mathbf{B}_{k-1}$. Hence, by statement II, we have $\mathbf{B}'_k = \mathbf{B}_k$. Therefore, the regions $\mathbf{B}^x_k$ and $\mathbf{B}^y_k$ also belong to the partition $\mathbf{B}'_k$. Consequently, since x and y are in distinct regions in the partition $\mathbf{B}'_k$, we conclude that u_{k+1} is a building edge for $\prec'$. Therefore, the set $E_\prec$ of building edges for $\prec$ is equal to the set $E_{\prec'}$ of building edges for $\prec'$.

Moreover, we conclude that $\mathbf{B}'_{k+1} = \mathbf{B}_{k+1}$ because both partitions result from the union of the two distinct regions of $\mathbf{B}_{k-1}$ containing x and y. Hence, for any $i > k+1$, as the edge of rank i for $\prec$ is also the edge of rank i for $\prec'$, we can conclude that any partition $\mathbf{B}_i$ is equal to the partition $\mathbf{B}'_i$. Hence, $\mathcal{B}_\prec$ and $\mathcal{B}_\prec$ have the same set of regions.

Since $E_\prec = E_{\prec'}$ and since $\mathcal{B}_\prec$ and $\mathcal{B}_\prec$ have the same set of regions, by Lemma 63, f is one-side increasing for $\prec'$. □

Lemma 68. *Let $\prec$ be an altitude ordering for w and let f be a saliency map such that f is one-side increasing for $\prec$. Let $(u_1, \ldots, u_{|E|})$ be the sequence induced by $\prec$. Let k be a critical rank for f and $\prec$ such that u_k is a building edge for $\prec$ and such that u_{k+1} is not a building edge for $\prec$. Then f is one-side increasing for the switch $\prec'$ for k.*

Proof. Let $(\mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_{|E|})$ be the sequence of partitions (of V) such that, for any i in $\{1, \ldots, |E|\}$, the partition $\mathbf{B}_i$ is the i-partition by the ordering $\prec$. Let $(\mathbf{B}'_0, \mathbf{B}'_1, \ldots, \mathbf{B}'_{|E|})$ be the sequence of partitions such that, for any i in $\{1, \ldots, |E|\}$, the partition $\mathbf{B}'_i$ is the i-partition by the ordering $\prec'$. As the sequences induced by $\prec$ and by $\prec'$ are equal for any edge with rank $i < k$, we may affirm that:

I. $\mathbf{B}_i = \mathbf{B}'_i$ for any $i < k$

Since u_k is a building edge for $\prec$, we have that:

II. $\mathbf{B}_k$ is different from $\mathbf{B}_{k-1}$

Let $u_k = \{s, r\}$ and $u_{k+1} = \{x, y\}$. Since $\mathbf{B}_k \neq \mathbf{B}_{k-1}$, we conclude that s and r are in distinct regions of $\mathbf{B}_{k-1}$. As u_{k+1} is not a building edge for $\prec$, we consider two cases: (1) x and y belong to a unique region of $\mathbf{B}_{k-1}$; and (2) x and y belong to two distinct regions of $\mathbf{B}_{k-1}$.

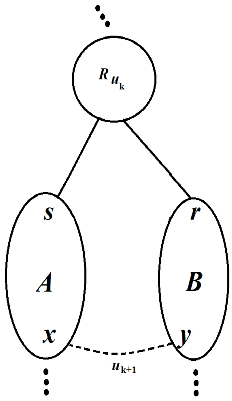
(1) Let us consider that x and y belong to a unique region of $\mathbf{B}_{k-1}$. By the statement I, we have $\mathbf{B}'_{k-1} = \mathbf{B}_{k-1}$. Hence, x and y belong to a unique region of $\mathbf{B}'_{k-1}$ and, therefore, u_{k+1} is not a building edge for $\prec'$. We will now prove that u_k is a building edge for $\prec'$. Since u_k is a building edge for $\prec$, we have that s and r belong to two distinct regions of the partition $\mathbf{B}_{k-1}$. Since u_{k+1} is not a building edge for $\prec'$, we have $\mathbf{B}'_k = \mathbf{B}'_{k-1}$. Then, by the statement I, we have $\mathbf{B}'_k = \mathbf{B}'_{k-1} = \mathbf{B}_{k-1}$. Therefore, s and r belong to two distinct regions of the partition $\mathbf{B}'_k$. Hence, u_k is a building edge for $\prec'$.

Therefore, the set $E_\prec$ of building edges for $\prec$ is equal to the set $E_{\prec'}$ of building edges for $\prec'$.

Moreover, we conclude that $\mathbf{B}'_{k+1} = \mathbf{B}_{k+1}$ because both partitions result from the union of the two distinct regions of $\mathbf{B}_{k-1}$ containing s and r. Hence, for any $i > k+1$, as the edge of rank i for $\prec$ is also the edge of rank i for $\prec'$, we can conclude that any partition $\mathbf{B}_i$ is equal to the partition $\mathbf{B}'_i$. Thus, $\mathcal{B}_\prec$ and $\mathcal{B}_\prec$ have the same set of regions.

Since $E_\prec = E_{\prec'}$ and since $\mathcal{B}_\prec$ and $\mathcal{B}_\prec$ have the same set of regions, by Lemma 63, f is one-side increasing for $\prec'$.

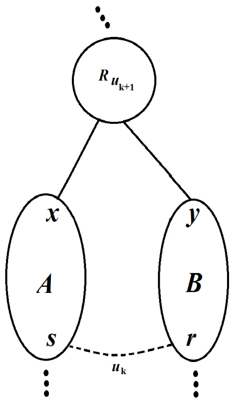
(2) We now consider that x and y belong to two distinct regions of $\mathbf{B}_{k-1}$. Let A and B be the regions of $\mathbf{B}_{k-1}$ such that $s \in A$ and $r \in B$. Since x and y belong to two distinct regions of $\mathbf{B}_{k-1}$ and since $\mathbf{B}_k = \{A \cup B\} \cup (\mathbf{B}_{k-1} \setminus \{A, B\})$, we conclude that either x or y is in A, and that either s or r is in B. Without loss of generality, let us assume that $x \in A$ and $y \in B$. This situation is illustrated in the following figure.



$\mathcal{B}_\prec$

Since u_{k+1} is the edge of rank k for the ordering $\prec'$, we can say that the k-partition $\mathbf{B}'_k$ by the ordering $\prec'$ is $\{A \cup B\} \cup (\mathbf{B}'_{k-1} \setminus \{A, B\})$ because A and B are the regions of $\mathbf{B}'_{k-1}$ that contain respectively x and y. As the region $\{A \cup B\}$ do not belong to the partition $\mathbf{B}'_{k-1}$, we have that u_{k+1} is the building edge of the region $\{A \cup B\}$. Hence, u_{k+1} is a building edge for $\prec'$.

Since u_k is the edge of rank $k+1$ for the ordering $\prec'$, we may conclude that $\mathbf{B}'_{k+1} = \mathbf{B}'_k$ because the s and r belong to the same region $\{A \cup B\}$ of $\mathbf{B}'_k$. Therefore, u_k is not a building edge for $\prec'$. This situation is illustrated in the following image.



$\mathcal{B}_{\prec'}$

We conclude that $\mathcal{B}_\prec$ and $\mathcal{B}_{\prec'}$ have the same set of regions but not the same set of building edges: $E_{\prec'} = E_\prec \setminus \{u_k\} \cup \{u_{k+1}\}$. Hence, the only difference between the hierarchies $\mathcal{B}_\prec$ and $\mathcal{B}_{\prec'}$ is the building edge of the region $\{A \cup B\}$. Therefore, we may say that, if the weight of the building edge of $\{A \cup B\}$ for $\prec$ is equal to the weight of the building edge of $\{A \cup B\}$ for $\prec'$, then f is also one-side increasing for $\prec'$. To that end, we will prove that $f(u_k) = f(u_{k+1})$.

By Lemma 84, as f is one-side increasing for $\prec$, we have that:

III. $(V, E_\prec)$ is a MST of (G, f)

By the statement III and by Lemma 83, we conclude that:

IV. the hierarchy $\mathcal{QFZ}(G, f)$ is equal to the hierarchy $\mathcal{QFZ}((V, E_\prec), f)$

Statement IV implies that f is the saliency map of the hierarchy $\mathcal{QFZ}((V, E_\prec), f)$. Hence, for any edge $u = \{a, b\}$ in E, $f(u)$ is the maximum weight in the unique path between a and b in $((V, E_\prec), f)$. We can affirm that:

V. the unique path between x and y in $((V, E_\prec), f)$ is a path that includes the edge u_k

By the statement V and by the definition of saliency maps, we have $f(u_{k+1}) \geq f(u_k)$. Since k is a critical rank for f and $\prec$, we have $f(u_{k+1}) \leq f(u_k)$. Therefore, we have $f(u_k) = f(u_{k+1})$, which completes the proof that f is one-side increasing for $\prec'$.

□

8.1.3 Proof of Property 22

(Property 22). *Let $\mathcal{H}$ be a hierarchy on V. The hierarchy $\mathcal{H}$ is a flattened hierarchical watershed of (G, w) if and only if there is an altitude ordering $\prec$ for w such that:*

1. *$(V, E_\prec)$ is a MST of $(G, \Phi(\mathcal{H}))$; and*

2. *for any edge u in $E_\prec$, if u is not a watershed-cut edge for $\prec$, then $\Phi(\mathcal{H})(u) = 0$; and*

3. *for any edge u in $E_\prec$, there exists a child R of R_u such that $\Phi(\mathcal{H})(u) \geq \vee\{\Phi(\mathcal{H})(v)$ such that R_v is included in $R\}$, where $\vee\{\} = 0$.*

To prove Property 22, we establish the following lemma.

Lemma 69. *Let $\prec$ be an altitude ordering for w and let $\mathcal{H}$ be a hierarchy on V such that $\Phi(\mathcal{H})$ is one-side increasing for $\prec$. Then $(V, E_\prec)$ is a MST of $(G, \Phi(\mathcal{H}))$.*

Proof. Let α denote the sum of the weight of the edges in $E_\prec$ in the map $\Phi(\mathcal{H})$: $\alpha = \sum_{e \in E_\prec} \Phi(\mathcal{H})(e)$. As $\Phi(\mathcal{H})$ is one-side increasing for $\prec$, by the condition 1 of Definition 18, we can affirm that $\alpha = 0 + 1 + \cdots + n - 1$. In order to prove that $(V, E_\prec)$ is a MST of $(G, \Phi(\mathcal{H}))$, we will prove that, for any MST G' of $(G, \Phi(\mathcal{H}))$, the sum of the weight of the edges in G' is greater than or equal to α. Let G' be a MST of $(G, \Phi(\mathcal{H}))$. As G' is a MST of $(G, \Phi(\mathcal{H}))$, by the condition 1 of Lemma 83, we have that G and G' have the same quasi-flat zones hierarchy: $\mathcal{QFZ}(G, \Phi(\mathcal{H})) = \mathcal{QFZ}(G', \Phi(\mathcal{H}))$. As $\Phi(\mathcal{H})$ is the saliency map of $\mathcal{H}$, we have that $\mathcal{H} = \mathcal{QFZ}(G, \Phi(\mathcal{H}))$. Therefore, $\mathcal{H} = \mathcal{QFZ}(G', \Phi(\mathcal{H}))$. Let i be a value in $\{1, \ldots, n-1\}$. By the condition 1 of Definition 18, we can say that $\{1, \ldots, n-1\}$ is a subset of the range of $\Phi(\mathcal{H})$. Therefore, $\mathcal{H}$ is composed of at least n distinct partitions. Let $\mathcal{H}$ be the sequence $(\mathbf{P}_0, \ldots, \mathbf{P}_{n-1}, \ldots)$. Since the partitions $\mathbf{P}_i$ and $\mathbf{P}_{i-1}$ are distinct, then there exists a region in $\mathbf{P}_i$ which is not in $\mathbf{P}_{i-1}$. Therefore, there is a region X of $\mathbf{P}_i$ which is composed of a several regions $\{R_1, R_2, \ldots\}$ of $\mathbf{P}_{i-1}$. Then, there are two adjacent vertices x and y such that x and y are in distinct regions in $\{R_1, R_2, \ldots\}$. Let x and y be two adjacent vertices such that x and y are in distinct regions in $\{R_1, R_2, \ldots\}$. Hence, the lowest j such that x and y belong to the same region of $\mathbf{P}_j$ is i. Thus, there exists an edge $u = \{x, y\}$ in $E_\prec$ such that $\Phi(\mathcal{H})(u) = i$. Hence, the sum of the weight of the edges of G' is at least $1 + \cdots + n - 1$, which is equal to α. Therefore, the graph $(V, E_\prec)$ is a MST of $(G, \Phi(\mathcal{H}))$. □

The reader can observe that the statement 3 of the above property is precisely the statement 3 of the definition of one-side increasing maps (Definition 18), and that the statement 2 is an implication of the statement 2 of Definition 18. The statement 1 of the above property corresponds to a property of one-side increasing maps established in Lemma 69.

In order to prove Property 22, we establish some auxiliary lemmas on MSTs and saliency maps.

In the following, we state a well-known property of spanning trees in Lemma 111.

Let x and y be two vertices in V and let $\pi = (x_0, \ldots, x_p)$ be a path from x to y. For any edge $u = \{x_{i-1}, x_i\}$ for i in $\{1, \ldots, p\}$, we say that u *is in* π or that π *includes* u.

Lemma 70. *Let G' be a spanning tree of a weighted graph (G, f). Let $u = \{x, y\}$ be an edge in $E \setminus E(G')$ and let π be the path from x to y (resp. y to x) in G'. The graph G' is a MST of (G, f) if and only if $f(u) \geq f(v)$ for any edge v in π.*

The following lemma characterizes MSTs of saliency maps.

Lemma 71. *Let f be the saliency map of a hierarchy on V and let G' be a spanning tree of (G, f). Let $u = \{x, y\}$ be an edge in $E \setminus E(G')$ and let π be the path from x to y (resp. y to x) in G'. Let v be an edge of greatest weight in π. The graph G' is a MST of (G, f) if and only if $f(u) = f(v)$.*

Proof. We will first prove the forward implication of this lemma. Let G' be a MST of $(G, \Phi(\mathcal{H}))$. Then, by Lemma 111, for any edge e in the path π, we have $\Phi(\mathcal{H})(e) \leq \Phi(\mathcal{H}(u)$. Hence, $\Phi(\mathcal{H})(v) \leq \Phi(\mathcal{H}(u)$. Let us assume that $\Phi(\mathcal{H})(v) < \Phi(\mathcal{H})(u)$. Then, given $\lambda = \Phi(\mathcal{H})(v)$, in the λ-level set of $(G, \Phi(\mathcal{H}))$, the vertices x and y are connected, which implies that, by the definition of saliency maps, $\Phi(\mathcal{H}(u)$ is less or equal to $\Phi(\mathcal{H})(v)$, which contradicts our assumption. Hence, $\Phi(\mathcal{H})(v) = \Phi(\mathcal{H}(u)$.

Now, let us assume that $\Phi(\mathcal{H})(u)$ is equal to the greatest weight among the edges in π. Then, for any edge e in the path π, we have $\Phi(\mathcal{H})(e) \leq \Phi(\mathcal{H}(u)$. Then, by Lemma 111, G' is a MST of $(G, \Phi(\mathcal{H}))$. □

Lemma 72. *Let $\mathcal{H}$ be a hierarchy on V and let $\mathcal{H}'$ be a flattening of $\mathcal{H}$. Let u and v be two distinct edges in E such that $\Phi(\mathcal{H}')(u) < \Phi(\mathcal{H}')(v)$. Then $\Phi(\mathcal{H})(u) < \Phi(\mathcal{H})(v)$.*

Proof. Let $u = \{x_1, y_1\}$ and $v = \{x_2, y_2\}$. As $\Phi(\mathcal{H}')(u) < \Phi(\mathcal{H}')(v)$, there is a partition $\mathbf{P}$ of $\mathcal{H}'$ such that x_1 and y_1 belong to the same region of $\mathbf{P}$ and we such that x_2 and y_2 do not belong to the same region of $\mathbf{P}$. As $\mathbf{P}$ is a partition of $\mathcal{H}'$, there is a partition in $\mathcal{H}$ such that x_1 and y_1 belong to the same region of this partition but x_2 and y_2 do not. Then, $\Phi(\mathcal{H})(u) < \Phi(\mathcal{H})(v)$. □

Lemma 73. *Let $\mathcal{H}$ be a hierarchy on V and let $\mathcal{H}'$ be a flattening of $\mathcal{H}$. Let u and v be two distinct edges in E such that $\Phi(\mathcal{H})(u) \leq \Phi(\mathcal{H})(v)$. Then $\Phi(\mathcal{H}')(u) \leq \Phi(\mathcal{H}')(v)$.*

Proof. Let $u = \{x_1, y_1\}$ and $v = \{x_2, y_2\}$. As $\Phi(\mathcal{H})(u) \leq \Phi(\mathcal{H})(v)$, then for any partition $\mathbf{P}$ of $\mathcal{H}$, if x_2 and y_2 are in the same region of $\mathbf{P}$, then x_1 and y_1 are in the same region of $\mathbf{P}$ as well. As any partition of $\mathcal{H}'$ is also a partition of $\mathcal{H}$, we may say that for any partition $\mathbf{P}$ of $\mathcal{H}'$, if x_2 and y_2 are in the same region of $\mathbf{P}$, then x_1 and y_1 are in the same region of $\mathbf{P}$. Hence, $\Phi(\mathcal{H}')(u) \leq \Phi(\mathcal{H}')(v)$. □

The forward and backward implications of Property 22 are proven in Lemmas 74 and 75, respectively.

Lemma 74. *Let $\mathcal{H}$ be a flattened hierarchical watershed of (G, w). Then, there is an altitude ordering $\prec$ for w such that:*

1. *$(V, E_\prec)$ is a MST of $(G, \Phi(\mathcal{H}))$; and*

2. *for any building edge u for $\prec$, if u is not a watershed-cut edge for $\prec$, then $\Phi(\mathcal{H})(u) = 0$; and*

3. *for any building edge u for $\prec$, there exists a child R of R_u such that $\Phi(\mathcal{H})(u) \geq \vee\{\Phi(\mathcal{H})(v)$ such that R_v is included in $R\}$, where $\vee\{\} = 0$.*

Proof. As $\mathcal{H}$ is a flattened hierarchical watershed of (G, w), by Definition 21, there is a hierarchical watershed $\mathcal{H}_w$ of (G, w) such that $\mathcal{H}$ is a flattening of $\mathcal{H}_w$. By Theorem 19, there is an altitude ordering $\prec$ for w such that $\Phi(\mathcal{H}_w)$ is one-side increasing for $\prec$. Let $\prec$ be the altitude ordering for w such that $\Phi(\mathcal{H}_w)$ is one-side increasing for $\prec$. By Lemma 84, $(V, E_\prec)$ is a MST of $(G, \Phi(\mathcal{H}_w))$. Let G' denote the graph $(V, E_\prec)$. By Lemma 83, $\mathcal{H}_w$ is the hierarchy $\mathcal{QFZ}(G', \Phi(\mathcal{H}_w))$. Then, any partition of $\mathcal{H}$ is a partition of $\mathcal{QFZ}(G', \Phi(\mathcal{H}_w))$. By the definition of saliency maps, we can affirm that any partition of $\mathcal{QFZ}(G, \Phi(\mathcal{H}))$ is a partition of $\mathcal{QFZ}(G', \Phi(\mathcal{H}_w))$.

In the following, we will prove that the three statements hold true for $\prec$.

1. We will first prove that G' is a MST of $(G, \Phi(\mathcal{H}))$. By contradiction, let us assume that G' is not a MST of $(G, \Phi(\mathcal{H}))$. Then, by Lemma 71, there is an edge $u = \{x, y\}$ such that u is in $E \setminus E(G')$ and such that $\Phi(\mathcal{H})(u)$ is different from the greatest weight among the edges in the path π from x to y in $(G', \Phi(\mathcal{H}))$. Let v be an edge of greatest weight in π. As $\mathcal{H}$ is equal to $\mathcal{QFZ}(G, \Phi(\mathcal{H}))$, we may affirm that $\Phi(\mathcal{H})(u)$ is lower than $\Phi(\mathcal{H})(v)$ because, otherwise, the vertices x and y would be connected in the λ-level set of $(G, \Phi(\mathcal{H}))$ for a λ lower than $\Phi(\mathcal{H})(u)$, which contradicts the fact that $\Phi(\mathcal{H})$ is a saliency map. Hence, we have $\Phi(\mathcal{H})(u) < \Phi(\mathcal{H})(v)$. Then, by Lemma 73, as $\mathcal{H}$ is a flattening of $\mathcal{H}_w$, we may conclude that $\Phi(\mathcal{H}_w)(u) < \Phi(\mathcal{H}_w)(v)$. Hence, the weight $\Phi(\mathcal{H}_w)(u)$ is different from the greatest weight among the edges in the path π. Therefore, by Lemma 71, G' is not a MST of $(G, \Phi(\mathcal{H}_w))$, which contradicts our assumption. Hence, we may conclude that G' is a MST of $(G, \Phi(\mathcal{H}))$.

2. We will now prove the second condition for $\mathcal{H}$ to be a flattened hierarchical watershed of (G, w). As $\mathcal{H}_w$ is one-side increasing for $\prec$, by the second condition of Definition 18, for any watershed-cut edge $u = \{x, y\}$ for $\prec$, we have $\Phi(\mathcal{H}_w)(u) = 0$. Then, for any partition $\mathbf{P}$ of $\mathcal{H}_w$, x and y belong to the same region of $\mathbf{P}$. Therefore, as any partition of $\mathcal{H}$ is a partition of $\mathcal{H}_w$, we can say that, for any partition $\mathbf{P}$ of $\mathcal{H}$, x and y belong to the same region of $\mathbf{P}$. Hence, the lowest λ such that x and y are the same partition $\mathbf{P}_\lambda$ of $\mathcal{H}$ is zero. Hence, $\Phi(\mathcal{H})(u) = 0$.

3. We will now prove the third condition for $\mathcal{H}$ to be a flattened hierarchical watershed of (G, w). By the third statement of Definition 18, we have that, for any edge u in

$E_\prec$, there exists a child R of R_u such that $\Phi(\mathcal{H}_w)(u) \geq \vee\{\Phi(\mathcal{H}_w)(v) \mid R_v \subseteq R\}$. Let u be an edge in $E_\prec$ and let R be the child of R_u such that $\Phi(\mathcal{H}_w)(u) \geq \vee\{\Phi(\mathcal{H}_w)(v) \mid R_v \subseteq R\}$. Let v be an edge in $E_\prec$ such that $R_v \subseteq R$. Then, $\Phi(\mathcal{H}_w)(u) \geq \Phi(\mathcal{H}_w)(v)$. Hence, by Lemma 73, $\Phi(\mathcal{H})(u) \geq \Phi(\mathcal{H})(v)$. Therefore, we may conclude that $\Phi(\mathcal{H})(u) \geq \vee\{\Phi(\mathcal{H})(v) \mid R_v \subseteq R\}$.

□

The following lemma corresponds to the backward implication of Property 22.

Lemma 75. *Let $\mathcal{H}$ be a hierarchy on V and let $\prec$ be an altitude ordering for w such that:*

1. *$(V, E_\prec)$ is a MST of $(G, \Phi(\mathcal{H}))$; and*
2. *for any edge u in $E_\prec$, if u is not a watershed-cut edge for $\prec$, then $\Phi(\mathcal{H})(u) = 0$; and*
3. *for any edge u in $E_\prec$, there exists a child R of R_u such that $\Phi(\mathcal{H})(u) \geq \vee\{\Phi(\mathcal{H})(v)$ such that R_v is included in $R\}$, where $\vee\{\} = 0$.*

Then $\mathcal{H}$ is a flattened hierarchical watershed of (G, w).

In order to prove Lemma 75, we first state two auxiliary lemmas. From Property 10 of [27], we can deduce the following lemma linking binary partition hierarchies and MSTs.

Lemma 76. *Let $\mathcal{B}$ be a binary partition hierarchy of (G, w). The graph $(V, E_\prec)$ is a MST of (G, w).*

By Property 12 of [27] linking hierarchical watersheds and hierarchies induced by an altitude ordering and a sequence of minima, and by Lemma 83, we infer the following lemma.

Lemma 77. *Let G' be a MST of (G, w) and let $\mathcal{H}$ be a hierarchical watershed of (G', w). Then $\mathcal{H}$ is also a hierarchical watershed of (G, w).*

of Lemma 75. Let $\mathcal{H}$ be a hierarchy on V such that there is an altitude ordering $\prec$ for w such that:

1. $(V, E_\prec)$ is a MST of $(G, \Phi(\mathcal{H}))$; and
2. for edge u in $E_\prec$, if u is not a watershed-cut edge for $\prec$, then $\Phi(\mathcal{H})(u) = 0$; and

3. for edge u in $E_\prec$, there exists a child R of R_u such that $\Phi(\mathcal{H})(u) \geq \vee\{\Phi(\mathcal{H})(v)$ such that R_v is included in $R\}$, where $\vee\{\} = 0$.

We will prove that $\mathcal{H}$ is a flattened hierarchical watershed of (G, w). To this end, we will prove that there is a hierarchical watershed $\mathcal{H}_w$ of (G, w) such that any partition of $\mathcal{H}$ is also a partition of $\mathcal{H}_w$. Let G' denote the graph $(V, E_\prec)$. By Lemma 76, G' is a MST of (G, w). Moreover, by Lemma 77, given a hierarchical watershed $\mathcal{H}_w$ of a MST of (G, w), we can say that $\mathcal{H}_w$ is also a hierarchical watershed of (G, w). Hence, we can simply prove that there is a hierarchical watershed $\mathcal{H}_w$ of (G', w) such that any partition of $\mathcal{H}$ is also a partition of $\mathcal{H}_w$.

To define the hierarchy $\mathcal{H}_w$, we first define a map f from $E_\prec$ into $\mathbb{R}$ such that f is one-side increasing for $\prec$. Since G' is a tree, by the definition of saliency maps, we can say that f is the saliency map of the hierarchy $\mathcal{QFZ}(G', f)$. By Theorem 19, as f is one-side increasing for $\prec$, we may say that $\mathcal{QFZ}(G', f)$ is a hierarchical watershed of (G', w).

In the map f, the edges which are not watershed-cut edges for $\prec$ are assigned to zero, and the watershed-cut edges for $\prec$ are ranked according to their weights in w and in $\Phi(\mathcal{H})$. Let $\prec_2$ be a total ordering on the set $\{u$ is a watershed-cut edge for $\prec\}$ such that, for any two watershed-cut edges u and v for $\prec$, we have $u \prec_2 v$ if and only if $\Phi(\mathcal{H})(u) < \Phi(\mathcal{H})(v)$ or if $\Phi(\mathcal{H})(u) = \Phi(\mathcal{H})(v)$ and $u \prec v$. The map f is defined as follows:

$$f(u) = \begin{cases} 0 & if\ u\ is\ not\ a\ watershed-cut \\ & edge\ for\ \prec \\ rank\ of\ u\ for\ \prec_2 & otherwise \end{cases} \tag{8.1}$$

We first demonstrate that f is one-side increasing for $\prec$.

1. By the definition of f, as there are $n-1$ watershed-cut edges for $\prec$, we can say that, for any i in $\{1, \ldots, n-1\}$, there is a watershed-cut edge u for $\prec$ such that the rank of u for $\prec_2$ is i and, consequently, $f(u) = i$. On the other hand, as w has at least one minimum, there is at least one edge e in $E_\prec$ such that e is not a watershed-cut edge for $\prec$ and such that $f(e) = 0$. Hence, we have $\{f(e) \mid u \in E_\prec\} = \{0, \ldots, n-1\}$. Therefore, the statement 1 of Definition 18 holds true for f.

2. For any edge u, by the definition of f, $f(u)$ is non-zero if and only if u is not a watershed-cut edge for $\prec$, so the statement 2 of Definition 18 holds true for f.

3. Let u be a building edge for $\prec$. If u is not a watershed-cut edge for $\prec$, then there is a child X of R_u such that there is no minimum of w included in X. Hence, none of the building edges of the descendants of X is a watershed-cut edge for $\prec$. By the definition of f, we have $f(u) = 0$ and, for any edge v such that $R_v \subseteq X$, we have $f(v) = 0$. Hence, $f(u) \geq \vee\{f(v)$ such that R_v is included in $X\}$. Otherwise, let us assume that u is a watershed-cut edge for $\prec$. Then there is at least one minimum of w included in each child of R_u. By the hypothesis 3, there is a child X of R_u such that $\Phi(\mathcal{H})(u) \geq \vee\{\Phi(\mathcal{H})(v)$ such that R_v is included in $X\}$. Let X be the child of R_u such that $\Phi(\mathcal{H})(u) \geq \vee\{\Phi(\mathcal{H})(v)$ such that R_v is included in $X\}$. Let e be a building edge for $\prec$ such that $R_e \subseteq X$. If e is not a watershed-cut edge for $\prec$, then $f(e) = 0$ and, consequently, $f(u) > f(e)$. Otherwise, if e is a watershed-cut edge for $\prec$, then we have $\Phi(\mathcal{H})(u) \geq \Phi(\mathcal{H})(e)$ and $e \prec u$, which implies that $e \prec_2 u$. Consequently, by the definition of f, we have $f(u) > f(e)$. Therefore, $f(u) \geq \vee\{f(v)$ such that R_v is included in $X\}$. Then, the third condition of Definition 18 holds true for f.

Hence, f is one-side increasing for $\prec$ and, as stated previously, $\mathcal{QFZ}(G', f)$ is a hierarchical watershed of (G', w) (resp. (G, w)). Now, we only need to prove that any partition of $\mathcal{H}$ is a partition of $\mathcal{QFZ}(G', f)$. By the hypothesis 1, G' is a MST of $(G, \Phi(\mathcal{H}))$. Then, by Lemma 83, we can say that $\mathcal{H}$ is the QFZ hierarchy of $(G', \Phi(\mathcal{H}))$. We will prove that any partition of $\mathcal{QFZ}(G', \Phi(\mathcal{H}))$ is also a partition of $\mathcal{QFZ}(G', f)$.

Let the range of $\Phi(\mathcal{H})$ be the set $\{0, \ldots, \ell\}$: $\{\Phi(\mathcal{H})(u) \mid u \in E_\prec\} = \{0, \ldots, \ell\}$. Let λ be a value in $\{0, \ldots, \ell\}$. Let $G'_{\lambda,\Phi(\mathcal{H})}$ be the λ-level set of $(G', \Phi(\mathcal{H}))$. Let α be the greatest value in $\{f(u) \mid u \in E(G'_{\lambda,\Phi(\mathcal{H})})\}$. We will prove that the α-level set of (G', f) is equal to the λ-level set of $(G', \Phi(\mathcal{H}))$. Since α is the greatest value in the set $\{f(u) \mid u \in E(G'_{\lambda,\Phi(\mathcal{H})})\}$, we can see that any edge v in the λ-level set of $(G', \Phi(\mathcal{H}))$ also belongs to the α-level set of (G', f). Now, we also need to prove that there is no edge u in the α-level set of (G', f) such that u is not in the λ-level set of $(G', \Phi(\mathcal{H}))$.

Let u be an edge which is not in the λ-level set of $(G', \Phi(\mathcal{H}))$. Then, $\Phi(\mathcal{H})(u) > \lambda$ and, for any edge v in the λ-level set of $(G', \Phi(\mathcal{H}))$, we have $\Phi(\mathcal{H})(u) > \Phi(\mathcal{H})(v)$. Since the minimum value of λ is zero, we can say that $\Phi(\mathcal{H})(u) > 0$ and, by the hypothesis 2, u is a watershed-cut edge for $\prec$. Let v be an edge in the λ-level set of $(G', \Phi(\mathcal{H}))$. Since $\Phi(\mathcal{H})(u) > \Phi(\mathcal{H})(v)$, if v is a watershed-cut edge for $\prec$, then $v \prec_2 u$ and $f(u) > f(v)$. Otherwise, if v is not a watershed-cut edge for $\prec$, by the definition of f, we have $f(v) = 0$ and $f(u) > f(v)$. Thus, for any edge v in the λ-level set of $(G', \Phi(\mathcal{H}))$, we have $f(u) > f(v)$ and, therefore, $f(u) > \alpha$. Then, u is not in the α-level set of (G', f).

Therefore, we can conclude that the α-level set of (G', f) is equal to the λ-level set of $(G', \Phi(\mathcal{H}))$. As the partitions of $\mathcal{H}$ are given by the set of connected components of the level sets of $(G', \Phi(\mathcal{H}))$, we can affirm that any partition of $\mathcal{H}$ is also a partition of $\mathcal{QFZ}(G', f)$. Therefore, there is a hierarchical watershed $\mathcal{H}_w = \mathcal{QFZ}(G', f)$ of (G', w) (resp. (G, w)) such that any partition of $\mathcal{H}$ is also a partition of $\mathcal{H}_w$. Then, $\mathcal{H}$ is a flattened hierarchical watershed of (G', w) (resp. (G, w)).

□

8.2 Proofs of theorem and properties of Chapter 4

8.2.1 Proof of Property 23

(Property 23). *Let $\prec$ be an altitude ordering for w and let ϵ be a map from the regions of $\mathcal{B}_\prec$ into $\mathbb{R}$. The map ϵ is an* extinction map for $\prec$ *if and only if the following statements hold true:*

- $\{\epsilon(R) \mid R$ *is a region of* $\mathcal{B}_\prec\} = \{0, \ldots, n\}$;
- *for any two distinct minima M_1 and M_2 of w, we have $\epsilon(M_1) \neq \epsilon(M_2)$; and*
- *for any region R of $\mathcal{B}_\prec$, we have that $\epsilon(R)$ is equal to $\vee\{\epsilon(M)$ such that M is a minimum of w included in $R\}$, where $\vee\{\} = 0$.*

We prove the forward and backward implications of Property 23 in Lemma 78 and Lemma 79, respectively.

Lemma 78. *Let $\prec$ be an altitude ordering for w and let ϵ be a map from the regions of $\mathcal{B}_\prec$ into $\mathbb{R}$. If the map ϵ is an* extinction map for $\prec$, *then the following statements hold true:*

1. $\{\epsilon(R) \mid R$ *is a region of* $\mathcal{B}_\prec\} = \{0, \ldots, n\}$;
2. *for any two distinct minima M_1 and M_2 of w, we have $\epsilon(M_1) \neq \epsilon(M_2)$; and*
3. *for any region R of $\mathcal{B}_\prec$, we have that $\epsilon(R)$ is equal to $\vee\{\epsilon(M)$ such that M is a minimum of w included in $R\}$, where $\vee\{\} = 0$.*

Proof. Let ϵ be an extinction map for $\prec$. Then, by the definition of extinction maps, there is a sequence $\mathcal{S} = (M_1, \ldots, M_n)$ of minima of w such that ϵ is the extinction map for $(\mathcal{S}, \prec)$. We will prove that the statements 1, 2 and 3 hold true for ϵ.

To prove that the statement 1 holds true, we will first prove that $\{\epsilon(R) \mid R$ is a region of $\mathcal{B}_\prec\} \subseteq \{0, \ldots, n\}$. Since w has n minima, the extinction value of any region of $\mathcal{B}_\prec$

which includes a minimum of w is in the set $\{1, \dots, n\}$. On the other hand, for any region R of $\mathcal{B}_\prec$ which do not include any minimum of w, we have that $\epsilon(R) = 0$. Hence, $\{\epsilon(R) \mid R$ is a region of $\mathcal{B}_\prec\} \subseteq \{0, \dots, n\}$. We will now prove that $\{0, \dots, n\} \subseteq \{\epsilon(R) \mid R$ is a region of $\mathcal{B}_\prec\}$. As $\mathcal{B}_\prec$ has at least one leaf-region composed of a single vertex of G, we can affirm that there is at least one region of $\mathcal{B}_\prec$ which do not include any minimum of w and whose extinction value for $(\mathcal{S}, \prec)$ is zero. Then, 0 is in $\{\epsilon(R) \mid R$ is a region of $\mathcal{B}_\prec\}$. Now, let i be a value in $\{1, \dots, n\}$. For the minimum M_i, we may affirm that M_i is the unique minimum of w included in M_i and, therefore, $\epsilon(M_i) = i$. Hence, i is in $\{\epsilon(R) \mid R$ is a region of $\mathcal{B}_\prec\}$. We may conclude that, for any i in $\{0, \dots, n\}$, i is in $\{\epsilon(R) \mid R$ is a region of $\mathcal{B}_\prec\}$. Therefore, the range of ϵ is $\{0, \dots, n\}$, which corresponds to the statement 1 of Lemma 78.

By the definition of extinction maps, for any minimum M_i, for i in $\{1, \dots, n\}$, we have $\epsilon(M_i) = i$ because M_i is the only minimum of w included in M_i. Therefore, for any two distinct minima M_i and M_j, for i, j in $\{1, \dots, n\}$, we have $\epsilon(M_i) = i$ and $\epsilon(M_j) = j$ and, consequently, $\epsilon(M_i)$ is different from $\epsilon(M_j)$. Hence, the statement 2 of Lemma 78 holds true for ϵ.

The statement 3 of Lemma 78 is precisely the definition of extinction values: for any region R of $\mathcal{B}_\prec$, the extinction value of R is zero if there is no minimum of w included in R and, otherwise, it is the maximal i (which is equal to $\epsilon(M_i)$) such that M_i is included in R. □

Lemma 79. *Let $\prec$ be an altitude ordering for w and let ϵ be a map from the regions of $\mathcal{B}_\prec$ into $\mathbb{R}$ such that:*

1. *$\{\epsilon(R) \mid R$ is a region of $\mathcal{B}_\prec\} = \{0, \dots, n\}$;*
2. *for any two distinct minima M_1 and M_2 of w, we have $\epsilon(M_1) \neq \epsilon(M_2)$; and*
3. *for any region R of $\mathcal{B}_\prec$, we have that $\epsilon(R)$ is equal to $\vee\{\epsilon(M)$ such that M is a minimum of w included in $R\}$, where $\vee\{\} = 0$.*

Then the map ϵ is an extinction map for $\prec$.

Proof. To prove that ϵ is an extinction map for $\prec$, we will show that there exists a sequence $S = (M_1, \dots, M_n)$ of minima of w such that, for any region R of $\mathcal{B}_\prec$, the value $\epsilon(R)$ is the extinction value of R for $(\mathcal{S}, \prec)$.

Let $\mathcal{S} = (M_1, \dots, M_n)$ be a sequence of minima of w ordered in non-decreasing order for ϵ, *i.e.*, for any two distinct minima M_i and M_j, with i, j in $\{1, \dots, n\}$, if $\epsilon(M_i) < \epsilon(M_j)$ then $i < j$.

By the hypothesis 2, this sequence $\mathcal{S}$ is unique. By the hypothesis 3, for any region R of $\mathcal{B}$ such that there is no minimum of w included in R, $\epsilon(R) = \vee\{\} = 0$, so $\epsilon(R)$ is the extinction value of R for $\prec$ and $\mathcal{S}$.

Since w has n minima, for any minimum M of w, the value $\epsilon(M)$ is in $\{1, \ldots, n\}$. Otherwise, by contradiction, let us assume that there exists a minimum M' of w such that $\epsilon(M') = 0$. Then, there is a value i in $\{1, \ldots, n\}$ such that, for any minimum M'' of w, the value $\epsilon(M'')$ is different from i. Consequently, by the hypothesis 3, the range of ϵ would be $\{0, \ldots, n\} \setminus \{i\}$, which contradicts the hypothesis 1. Therefore, for any minimum M_i of w, for i in $\{1, \ldots, n\}$, as our assumption that $\epsilon(M_i) < \epsilon(M_j)$ implies that $i < j$, we have that $\epsilon(M_i) = i$. Thus, $\epsilon(M_i)$ is the extinction value of M_i for $\prec$ and S.

It follows that, by the hypothesis 3, for any region R of $\mathcal{B}_\prec$ such that there is a minimum of w included in R, the value $\epsilon(R)$ is the maximum value i (which is equal to $\epsilon(M_i)$) in $\{1, \ldots, n\}$ such that M_i is included in R.

Thus, for any region R of $\mathcal{B}_\prec$, the value $\epsilon(R)$ is the extinction value of R for $(\mathcal{S}, \prec)$. Therefore, the map ϵ is an extinction map for $\prec$. □

8.2.2 Proof of Property 24

(Property 24). *Let $\mathcal{H}$ be a hierarchy on V. The hierarchy $\mathcal{H}$ is a hierarchical watershed of (G, w) if and only if there exists an altitude ordering $\prec$ for w and an extinction map ϵ for $\prec$ such that:*

1. *$(V, E_\prec)$ is a MST of $(G, \Phi(\mathcal{H}))$; and*
2. *for any edge u in $E_\prec$, the value $\Phi(\mathcal{H})(u)$ is equal to* $\min\{\epsilon(R)$ *such that R is a child of $R_u\}$.*

To prove Property 24, we first present some auxiliary lemmas. From the results established in [74], we can state the following lemma.

Lemma 80. *Let $\mathcal{B}$ be a binary partition hierarchy of (G, w). Then, any minimum of w is a region of $\mathcal{B}$.*

Lemma 81. *Let $\prec$ be an altitude ordering on the edges of G for w, let $\mathcal{S} = (M_1, \ldots, M_n)$ be a sequence of minima of w and let ρ be the persistence map for $(\mathcal{S}, \prec)$. The range of ρ is $\{0, \ldots, n-1\}$.*

Proof. Let ϵ be the extinction map for $(\mathcal{S}, \prec)$. We will prove that (1) for any building edge u for $\prec$, $\rho(u)$ is in $\{0, \ldots, n-1\}$, and that, (2) for any i in $\{0, \ldots, n-1\}$, there is a building edge u for $\prec$ such that $\rho(u) = i$.

1. $\{0, \ldots, n-1\} \subseteq range(\rho)$. First, we prove that 0 is in $range(\rho)$. By Property 23, there is a region X of $\mathcal{B}_\prec$ whose extinction value is zero. Therefore, the persistence value of the building edge u of the parent of X is equal to zero: $\rho(u) = 0$. Now, we will prove that any i in $\{1, \ldots, n-1\}$ is in $range(\rho)$. Let i be a value in $\{1, \ldots, n-1\}$. By Lemma 80, the minimum M_i is a region of $\mathcal{B}_\prec$. Then, there is a region of $\mathcal{B}_\prec$ whose extinction value is i. Let X be the largest region of $\mathcal{B}_\prec$ whose extinction value is i. We can say that $X \neq V$ because M_n is included in V and, therefore, $\epsilon(V) = n$. Let Z be the parent of X. We can infer that the extinction value $\epsilon(Z)$ of Z is strictly greater than i. Therefore, there is a minimum M_j with $j > i$ included in the sibling of X. Hence, the extinction value of $sibling(X)$ is also strictly greater than i. Then, the persistence value of the building edge of Z, being the minimum of the extinction value of its children, is i.

2. $range(\rho) \subseteq \{0, \ldots, n-1\}$. Let u be an edge in $E_\prec$. By Property 23 (statement 1), and as the persistence value of u is equal to the extinction value of a child of R_u, we have that $\rho(u)$ is in $\{0, \ldots, n\}$. Moreover, the persistence value $\rho(u)$ of u is lower than n because, if the extinction value of one child X of R_u is n, then the minimum M_n is included in X and M_n is not included in $sibling(X)$, which implies that the extinction value of $sibling(X)$ is strictly lower than n. Therefore, since $\rho(u) = min\{\epsilon(X), \epsilon(sibling(X))\}$, the persistence value of u is strictly lower than n. Thus, we have that $range(\rho) \subseteq \{0, \ldots, n-1\}$.

□

Lemma 82. *Let $\prec$ be an altitude ordering for w, let $\mathcal{S} = (M_1, \ldots, M_n)$ be a sequence of minima of w and let ρ be the persistence map for $(\mathcal{S}, \prec)$. Let $\mathcal{H}$ be the hierarchy induced by $\prec$ and $\mathcal{S}$. For any edge u in $E_\prec$, we have $\Phi(\mathcal{H})(u) = \rho(u)$.*

Proof. By Definition 5, the hierarchy $\mathcal{H}$ is the sequence $(CC(V, B_0), \ldots, CC(V, B_{n-1}))$ such that, for any i in $\{0, \ldots, n-1\}$, B_i is the set of building edges for $\prec$ whose persistence values are lower than or equal to i. Let $u = \{x, y\}$ be a building edge for $\prec$ and let i be the persistence value of u. We can say that x and y are in the same region of $CC(V, B_i)$ but in distinct regions of $CC(V, B_{i-1})$ if $i \neq 0$. Therefore, since $CC(V, B_i)$ is the i-th partition of $\mathcal{H}$, by the definition of saliency maps, we have $\Phi(\mathcal{H})(u) = i$. □

The following lemma, established in [25], links MSTs and QFZ hierarchies.

Lemma 83 (Theorem 4 of [25]). *A subgraph G' of G is a MST of (G, w) if and only if:*

1. *the QFZ hierarchy of G' and G are the same; and*

2. *the graph G' is minimal for statement 1,* i.e., *for any subgraph G'' of G', if the quasi-flat zone hierarchy of G'' for w is the one of G for w, then we have $G'' = G'$.*

Lemma 84. *Let $\prec$ be an altitude ordering for w and let $\mathcal{S} = (M_1, \dots, M_n)$ be a sequence of minima of w. Let $\mathcal{H}$ be the hierarchy induced by $\prec$ and $\mathcal{S}$. Then $(V, E_\prec)$ is a MST of $(G, \Phi(\mathcal{H}))$.*

Proof. Let α denote the sum of the weight of the edges in $E_\prec$ in the map $\Phi(\mathcal{H})$: $\alpha = \sum_{e \in E_\prec} \Phi(\mathcal{H})(e)$. Let ρ be the persistence map for $(\mathcal{S}, \prec)$. By Lemma 82, we can affirm that, for any edge u in $E_\prec$, we have $\Phi(\mathcal{H})(u) = \rho(u)$. Hence, we have $\alpha = \sum_{e \in E_\prec} \rho(e)$. We will first prove that α is precisely $0 + 1 + \dots + n - 1$. We know that, for any edge u in $E_\prec$:

1. if u is a watershed-cut edge for $\prec$, then each child of R_u contains at least one minimum of w. Therefore, the extinction values of both children of R_u is non-zero, and, consequently, the persistence value $\rho(u)$ of u is non-zero.

2. otherwise, if u is not a watershed-cut edge for $\prec$, then there exists a child X of R_u such that there is no minimum of w included in X. Therefore, the extinction value of X is zero. Since the extinction value of *sibling*(X) is at least zero by Lemma 78 (statement 1), the persistence value $\rho(u)$ of u, being the minimum between the extinction values of X and *sibling*(X), is also zero.

Hence, since there are $n-1$ watershed-cut edges for $\prec$, and since only the watershed-cut edges for $\prec$ have non-zero persistence values, we can conclude that, for any i in $\{1, \dots, n-1\}$, there is exactly one edge u in $E_\prec$ such that $\rho(u) = i$. Hence, $\alpha = \sum_{e \in E_\prec} \rho(e) = 0 + 1 + \dots + n - 1$.

Now, in order to prove that $(V, E_\prec)$ is a MST of $(G, \Phi(\mathcal{H}))$, we will prove that, for any MST G' of $(G, \Phi(\mathcal{H}))$, the sum of the weight of the edges in G' is greater than or equal to α. Let G' be a MST of $(G, \Phi(\mathcal{H}))$. As G' is a MST of $(G, \Phi(\mathcal{H}))$, by the condition 1 of Lemma 83, we have that G and G' have the same quasi-flat zones hierarchies: $\mathcal{QFZ}(G, \Phi(\mathcal{H})) = \mathcal{QFZ}(G', \Phi(\mathcal{H}))$. As $\Phi(\mathcal{H})$ is the saliency map of $\mathcal{H}$, we have that $\mathcal{H} = \mathcal{QFZ}(G, \Phi(\mathcal{H}))$. Therefore, $\mathcal{H} = \mathcal{QFZ}(G', \Phi(\mathcal{H}))$. Let i be a value in $\{1, \dots, n-1\}$. Since $\sum_{e \in E_\prec} \Phi(\mathcal{H})(e) = 0+1+\dots+n-1$, we can say that $\{1, \dots, n-1\}$ is a subset of the range of $\Phi(\mathcal{H})$. Therefore, $\mathcal{H}$ is composed of at least n distinct partitions. Let $\mathcal{H}$ be the sequence $(\mathbf{P}_0, \dots, \mathbf{P}_{n-1}, \dots)$. Since the partitions $\mathbf{P}_i$ and $\mathbf{P}_{i-1}$ are distinct, then there exists a region in $\mathbf{P}_i$ which is not in $\mathbf{P}_{i-1}$. Therefore, there is a region X of $\mathbf{P}_i$ which is composed of several regions $\{R_1, R_2, \dots\}$ of $\mathbf{P}_{i-1}$. Then, there are two adjacent vertices x and y such that x and y are in distinct regions in $\{R_1, R_2, \dots\}$. Let x and y be two adjacent vertices such that x and y are in distinct regions in $\{R_1, R_2, \dots\}$. Hence, the lowest j such that x and y belong to the same region of $\mathbf{P}_j$ is i. Thus, there exists an edge $u = \{x, y\}$ in $E_\prec$ such that $\Phi(\mathcal{H})(u) = i$. Hence, the sum of the weight of the edges of G' is at least $1 + \dots + n - 1$, which is equal to α. Therefore, the graph $(V, E_\prec)$ is a MST of $(G, \Phi(\mathcal{H}))$. □

Proof of Property 24. We first prove the forward implication of this property. Let $\mathcal{H}$ be a hierarchical watershed of (G, w). Then there is a sequence $\mathcal{S}$ of minima of w such that $\mathcal{H}$ is the hierarchical watershed of (G, w) for $\mathcal{S}$. Let $\mathcal{S}$ be the sequence of minima of w such that $\mathcal{H}$ is the hierarchical watershed of (G, w) for $\mathcal{S}$. By Property 6, there is an altitude ordering $\prec$ such that $\mathcal{H}$ is the hierarchy induced by $\prec$ and $\mathcal{S}$. Let $\prec$ be an altitude ordering such that $\mathcal{H}$ is the hierarchy induced by $\prec$ and $\mathcal{S}$. Then, by Lemma 84, $(V, E_\prec)$ is a MST of $(G, \Phi(\mathcal{H}))$. We will now prove the second statement of Property 24. By Lemma 82, for any edge u in $E_\prec$, $\Phi(\mathcal{H})(u)$ is equal to the persistence value $\rho(u)$ of u for $(\mathcal{S}, \prec)$. By the definition of persistence values, for edge u in $E_\prec$, the persistence value of u for $(\mathcal{S}, \prec)$ is the minimum extinction value of the children of R_u. Therefore, we can conclude that, for edge u in $E_\prec$, $\Phi(\mathcal{H})(u) = \min\{\epsilon(R)$ such that R is a child of $R_u\}$, where ϵ is the extinction map for $(\mathcal{S}, \prec)$. Hence, there exists an extinction map ϵ such that, for edge u in $E_\prec$, $\Phi(\mathcal{H})(u) = \min\{\epsilon(R)$ such that R is a child of $R_u\}$.

We will now prove the backward implication of Property 24. Let $\mathcal{H}$ be a hierarchy on V such that there exists an altitude ordering $\prec$ for w and an extinction map ϵ for $\prec$ such that:

1. $(V, E_\prec)$ is a MST of $(G, \Phi(\mathcal{H}))$; and
2. for any edge u in $E_\prec$, we have: $\Phi(\mathcal{H})(u) = \min\{\epsilon(R)$ such that R is a child of $R_u\}$.

Let G' denote the graph $(V, E_\prec)$. By Lemma 83 (statement 1), as G' is a MST of $(G, \Phi(\mathcal{H}))$, we have that G' and G have the same quasi-flat zones hierarchies (for $\Phi(\mathcal{H})$): $\mathcal{QFZ}(G', \Phi(\mathcal{H})) = \mathcal{QFZ}(G, \Phi(\mathcal{H}))$. Let ρ be the persistence map for $(\mathcal{S}, \prec)$. By the definition of persistence values, we can affirm that, for any edge u in $E_\prec$, we have $\Phi(\mathcal{H})(u) = \rho(u)$. Hence, we can say that $\mathcal{QFZ}(G', \Phi(\mathcal{H})) = \mathcal{QFZ}(G', \rho))$. Let $\mathcal{H}'$ be the hierarchy induced by $\prec$ and $\mathcal{S}$. By Lemma 84, G' is a MST of $(G, \Phi(\mathcal{H}'))$. Hence, by Lemma 83, G' and G have the same quasi-flat zones hierarchies (for $\Phi(\mathcal{H}')$): $\mathcal{QFZ}(G', \Phi(\mathcal{H}')) = \mathcal{QFZ}(G, \Phi(\mathcal{H}'))$. By Lemma 82, for edge u in $E_\prec$, we have $\Phi(\mathcal{H}')(u) = \rho(u)$, which is equal to $\Phi(\mathcal{H})(u)$ as stated previously. Thus, $\mathcal{QFZ}(G', \Phi(\mathcal{H}')) = \mathcal{QFZ}(G', \Phi(\mathcal{H}))$ and, consequently, $\mathcal{H}$ and $\mathcal{H}'$ are equal. By Property 6, $\mathcal{H}'$ is a hierarchical watershed of (G, w). Therefore, $\mathcal{H}$ is a hierarchical watershed of (G, w). □

8.2.3 Proof of Theorem 28

(**Theorem 28).** *Let f be a map from E into $\mathbb{R}$, let $\prec$ be a lexicographic ordering for (w, f), and let ξ be the approximated extinction map for $(f, \prec)$. The map f is the saliency map of a hierarchical watershed of (G, w) if and only if the map ξ is an extinction map for $\prec$.*

In order to prove Theorem 28, we establish lemmas 85 and 94.

Lemma 85. *Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$ such that f is one-side increasing for $\prec$. The approximated extinction map for $(f, \prec)$ is an extinction map for $\prec$.*

In order to prove Lemma 85, we prove in Lemmas 87, 88 and 92 that the three conditions of Property 23 for ξ to be an extinction map are satisfied. We first establish the following auxiliary lemma.

Lemma 86. *Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$ such that f is one-side increasing for $\prec$. Then, the two following statements hold true:*

1. *the set $\{f(e) \mid e\ is\ a\ watershed-cut\ edge\ for\ \prec\}$ is equal to $\{1, \ldots, n-1\}$; and*
2. *for any two distinct watershed-cut edges u and v for $\mathcal{B}$, we have $f(u) \neq f(v)$.*

Proof. By the Definition 18 (statement 1), we have $\{f(u) \mid u \in E_{\prec}\} = \{0, \ldots, n-1\}$ and, by Definition 18 (statement 2), only the weight of the watershed-cut edges for $\prec$ are strictly greater than zero. Then, $\{f(e) \mid e\ is\ a\ watershed-cut\ edge\ for\ \prec\} = \{1, \ldots, n-1\}$. Hence, for any i in $\{1, \ldots, n-1\}$, there is a watershed-cut edge e for $\prec$ such that $f(e) = i$. Moreover, as there are $n-1$ watershed-cut edges for $\prec$, for any two distinct watershed-cut edges u and v for $\prec$, we have $f(u) \neq f(v)$. □

Lemma 87. *Let $\prec$ be an altitude ordering for w, let f be a map from E into $\mathbb{R}$ such that f is one-side increasing for $\prec$, and let ξ be the approximated extinction map for $(f, \prec)$. The range of ξ is $\{0, \ldots, n\}$.*

Proof. We will prove that: (1) for any i in $\{0, \ldots, n\}$, there is a region R of $\mathcal{B}_{\prec}$ such that $\xi(R) = i$; and (2) for any region R of $\mathcal{B}_{\prec}$, we have $\xi(R)$ in $\{0, \ldots, n\}$.

(1) We first prove statement (1). We start by proving that there is a region R of $\mathcal{B}_{\prec}$ such that $\xi(R) = n$. Let R be the set V of vertices of G. Then, by Definition 27 (statement 1), we have $\xi(R) = k + 1$, where k is the supremum descendant

value of R for $(f, \prec)$. By Definition 18 (statement 1), we have $\{f(u) \mid u \in E_\prec\} = \{0, \ldots, n-1\}$. As $k = \vee\{f(u) \mid R_u \subseteq V\} = \vee\{0, \ldots, n-1\} = n-1$, we have that $\xi(R) = n - 1 + 1 = n$.

We will now show that there is a region R of $\mathcal{B}_\prec$ such that $\xi(R) = 0$. Let R be a region of $\mathcal{B}_\prec$ such that there is no minimum of w included in R. Then R is not a minimum of w and, consequently, the building edge of the parent of R is not a watershed-cut edge for $\prec$. Let u be building edge of the parent of R. Since there is no minimum of w included in R, by Definition 26, R is not a dominant region for $(f, \prec)$. By the statement 3 of the definition of approximated extinction maps (Definition 27), we have $\xi(R) = f(u)$. Since f is a one-side increasing map and since u is not a watershed-cut edge for $\prec$, we have $f(u) = 0$. Therefore, we have $\xi(R) = f(u) = 0$.

Finally, we will prove that, for any i in $\{1, \ldots, n-1\}$, there is a region R of $\mathcal{B}_\prec$ such that $\xi(R) = i$. By Lemma 86, we can say that, for any i in $\{1, \ldots, n-1\}$, there is a watershed-cut u edge for $\prec$ such that $f(u) = i$. Let u be a watershed-cut edge for $\prec$ and let X and Y be the children of R_u. Since u is a watershed-cut edge for $\prec$, both X and Y contain at least a minimum of w and, then, neither X nor Y are leaf regions of $\mathcal{B}_\prec$. Let $\ll$ be the non-leaf ordering for $(f, \prec)$. Since $\ll$ is a total ordering, we have either $X \ll Y$ or $Y \ll X$. Then, exactly one child of R_u is a dominant region for $(f, \prec)$. Let Y be the child of R_u which is not a dominant region for $(f, \prec)$. By Definition 27 (statement 3), we have $\xi(Y) = f(u)$. Therefore, for any i in $\{1, \ldots, n-1\}$, there is a watershed-cut edge u for $\prec$ such that $f(u) = i$ and such that there is a child Z of R_u such that $\xi(Z) = i$.

(2) We will now prove the statement (2). Let R be a region of $\mathcal{B}_\prec$. If $R = V$, then $\xi(R) = n$, as established in the proof of statement (1). Otherwise, let v be the building edge of the parent of R. By Definition 27, the value $\xi_f(R)$ is either $f(v)$ or $\xi(parent(R))$. Hence, either $\xi_f(R)$ is equal to $f(v)$ for a building edge v for $\prec$, or $\xi_f(R)$ is equal to $\xi(V) = n$. It is enough to prove that n and $f(v)$ are in $\{0, \ldots, n\}$. As f is one-side increasing for $\prec$, by Definition 18 (statement 1), we have $\{f(u) \mid u \in E_\prec\} = \{0, \ldots, n-1\}$. Since v is a building edge for $\prec$, we may say that $f(v)$ is in $\{0, \ldots, n-1\}$.

□

Lemma 88. *Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$ such that f is one-side increasing for $\prec$. Let ξ be the approximated extinction map for $(f, \prec)$. For any two minima M_1 and M_2 of w, if $\xi(M_1) = \xi(M_2)$, then $M_1 = M_2$.*

To prove Lemma 88, we first present the Lemmas 89, 90 and 91. In the following, for any non-leaf region X of a binary partition hierarchy $\mathcal{B}$ of (G, w), we denote by u_X the building edge of X.

Lemma 89. *Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$ such that f is one-side increasing for $\prec$. Let ξ be the approximated extinction map for $(f, \prec)$. For any region X of $\mathcal{B}_\prec$ such that there is a minimum M of w such that $M \subset X$, there is a child Y of X such that:*

1. $\xi(Y) = \xi(X)$;
2. $\xi(sibling(Y)) = f(u_X)$; *and*
3. *there is a minimum of w included in Y.*

Proof. Let X be a region such that there is a minimum M of w such that $M \subset X$. Then, there is a child Z of X such that there is a minimum M such that $M \subseteq Z$. Let Z be a child X such that there is a minimum M such that $M \subseteq Z$. We consider two cases: (1) $sibling(Z)$ is a leaf-region of $\mathcal{B}_\prec$; and (2) $sibling(Z)$ is a non-leaf region of $\mathcal{B}_\prec$.

(1) If $sibling(Z)$ is a leaf-region of $\mathcal{B}_\prec$, then, by Definition 26, Z is a dominant region for $(f, \prec)$ and $sibling(Z)$ is not a dominant region for $(f, \prec)$. Hence, by Definition 27, $\xi(Z) = \xi(X)$ and $\xi(sibling(Z)) = f(u_X)$.

(2) Let us now assume that $sibling(Z)$ is a non-leaf region of $\mathcal{B}_\prec$. Since X is not a minimum of w and since there is a minimum of w included in Z, we can conclude that there is a minimum of w included in $sibling(Z)$ as well. Let $\ll$ be the non-leaf ordering for $(f, \prec)$. As the non-leaf ordering $\ll$ is a total ordering on the non-leaf regions of $\mathcal{B}_\prec$, we have either $Z \ll sibling(Z)$ or $sibling(Z) \ll Z$. Then, by the definition of dominant regions (Definition 26), we have that either Z or $sibling(Z)$ is a dominant region for $(f, \prec)$. Let us assume that Z is a dominant region for $(f, \prec)$. Then, by Definition 27, we have $\xi(Z) = \xi(X)$ and $\xi(sibling(Z)) = f(u_X)$. Otherwise, if $sibling(Z)$ is a dominant region for $(f, \prec)$, we have $\xi(sibling(Z)) = \xi(X)$ and $\xi(Z) = f(u_X)$. Since both Z and $sibling(Z)$ include at least one minimum of w, we may say that there is a child Y of X for which the hypothesis 1, 2 and 3 hold true.

□

Lemma 90. *Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$ such that f is one-side increasing for $\prec$. Let ξ be the approximated extinction map for $(f, \prec)$. Let u be a watershed-cut edge for $\prec$. Then, there is a minimum M of w such that $\xi(M) = f(u)$.*

Proof. As u is a watershed-cut edge for $\prec$, each child of R_u includes at least one minimum of w. Then, there is a minimum M of w such that $M \subset R_u$. By Lemma 89, there is a child Y_1 of R_u such that $\xi(Y_1) = f(u)$. If Y_1 is a minimum of w, then the property holds true. Otherwise, if Y_1 is not a minimum of w, it means that there is a minimum M of w such that $M \subset Y_1$. By Lemma 89, there is a child Y_2 of Y_1 such that $\xi(Y_2) = \xi(Y_1) = f(u)$ and such that there is a minimum of w included in Y_2. Again, if Y_2 is a minimum of w, then the property holds true. Otherwise, we can apply this same reasoning indefinitely. We can define a sequence $(Y_1, \dots, Y_p)$ of regions of $\mathcal{B}_\prec$ where Y_p is a minimum of w and such that $\xi(Y_p) = \dots = \xi(Y_1) = f(u)$ and $Y_i \subset Y_{i-1}$ for any i in $\{2, \dots, p\}$. Therefore, there is a minimum Y_p included in R_u such that $\xi(Y_p) = f(u)$. □

Lemma 91. *Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$ such that f is one-side increasing for $\prec$. Let ξ be the approximated extinction map for $(f, \prec)$. Let X be a region of $\mathcal{B}_\prec$ such that X contains at least one minimum of w. There exists a minimum $M \subseteq X$ such that $\xi(M) = \xi(X)$.*

Proof. If X is a minimum of w, then it is trivial. Otherwise, by Lemma 89, there is a child Y_1 of X such that $\xi(Y_1) = \xi(X)$ and such that there is a minimum of w included in Y_1. If Y_1 is a minimum of w, then the property holds true. Otherwise, by Lemma 89, there is a child Y_2 of Y_1 such that $\xi(Y_2) = \xi(Y_1) = \xi(X)$ and such that there is a minimum of w included in Y_2. Again, if Y_2 is a minimum of w, then the property holds true. Otherwise, we can apply this same reasoning indefinitely. We can define a sequence $(Y_1, \dots, Y_p)$ of regions of $\mathcal{B}_\prec$ where Y_p is a minimum of w and such that $\xi(Y_p) = \dots = \xi(Y_1) = \xi(X)$ and $Y_i \subset Y_{i-1}$ for any i in $\{2, \dots, p\}$. Therefore, there is a minimum Y_p included in X such that $\xi(Y_p) = \xi(Y)$. □

Proof of Lemma 88. In order to prove that

(1) for any two minima M_1 and M_2 of w, if $\xi(M_1) = \xi(M_2)$, then $M_1 = M_2$,

we will prove that

(2) for any two minima M_1 and M_2 of w, we have $\xi(M_1) \neq \xi(M_2)$.

As w has n minima, it suffices to prove that, for any i in $\{1, \ldots, n\}$, there is a minimum M of w such that $\xi(M) = i$.

By Lemma 90, for any watershed-cut edge u for $\mathcal{B}_{\prec}$, there is a minimum M such that $\xi(M) = f(u)$. By Lemma 86, for any i in $\{1, \ldots, n-1\}$, there is a watershed-cut edge such that $f(u) = i$. Then, for any i in $\{1, \ldots, n-1\}$, there is a minimum M of w such that $\xi(M) = i$.

Since, f is one-side increasing for $\prec$, we have $\vee\{f(v) \mid R_v \in V\} = \{0, \ldots, n-1\}$. Then, we can conclude that $\xi(V) = \vee\{f(v) \mid R_v \in V\} + 1 = (n-1) + 1 = n$. By Lemma 91, there is a minimum M of w such that $\xi(M) = \xi(V) = n$.

Therefore, for any i in $\{1, \ldots, n\}$, there is a minimum M of w such that $\xi(M) = i$. Since w has n minima, it implies that the values $\xi(M_1)$ and $\xi(M_2)$ are distinct for any pair (M_1, M_2) of distinct minima of w. Hence, for any two minima M_1 and M_2 of w, if $\xi(M_1) = \xi(M_2)$, then $M_1 = M_2$. □

Lemma 92. *Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$ such that f is one-side increasing for $\prec$. Let ξ be the approximated extinction map for $(f, \prec)$. For any region R of $\mathcal{B}_{\prec}$, we have $\xi_f(R) = \vee\{\xi_f(M)$ such that M is a minimum of w included in $R\}$.*

To prove Lemma 92, we introduce Lemma 93.

Lemma 93. *Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$ such that f is one-side increasing for $\prec$. Let ξ be the approximated extinction map for $(f, \prec)$. Let X be a region of $\mathcal{B}_{\prec}$. Then $\xi(X)$ is greater than or equal to the supremum descendant value of X (for $(f, \prec)$).*

Proof. Let $\bigtriangledown$ be a map from each region of $\mathcal{B}_{\prec}$ into its supremum descendant value for $(f, \prec)$. We consider the following cases: (1) $X = V$, (2) $X \neq V$ and X is not a dominant region for $(f, \prec)$; and (3) X is a dominant region for $(f, \prec)$. Let $\ll$ be the non-leaf ordering for $(f, \prec)$.

1. If $X = V$, then $\xi(X) = \xi(V) = k+1$, where k is the supremum descendant value of X for $(f, \prec)$ (first case of Definition 27). Then, $\xi(X)$ is clearly than the supremum descendant value of X.

2. If $X \neq V$ and if X is not a dominant region for $(f, \prec)$, then $\xi(X) = f(u)$ (third case of Definition 27), where u is the building edge of the parent of X. By the definition of dominant regions, we consider two cases: (a) there is no minimum M of w such that $M \subseteq X$; or (b) $X \ll \mathit{sibling}(X)$.

(a) If there is no minimum M of w such that $M \subseteq X$, then there is no descendant of X whose building edge is a watershed-cut edge for $\prec$. Hence, for any edge v such that $R_v \subseteq X$, u is not a watershed-cut edge for $\prec$ and, since f is one-side increasing for $\prec$, we have $f(v) = 0$ Definition 18 (statement 2). Therefore, the supremum descendant value of X is zero. By Definition 18 (statement 1), we have $\{f(v) \mid v \in E_{\prec}\} = \{0, \ldots, n-1\}$. Hence, $\xi(X)$, being equal to $f(u)$, is greater than or equal to the supremum descendant value of X.

(b) If $X \ll \mathit{sibling}(X)$, then, by the definition of non-leaf ordering, we have:

i. either $\triangledown(X) < \triangledown(\mathit{sibling}(X))$; or

ii. $\triangledown(X) = \triangledown(\mathit{sibling}(X))$ and $u_X \prec u_{\mathit{sibling}(X)}$.

Thus, we have $\triangledown(X) \leq \triangledown(\mathit{sibling}(X))$. Since f is one-side increasing for $\prec$, by the statement 3 of Definition 18, there is a child Y of $\mathit{parent}(X)$ such that $f(u) \geq \vee\{f(v) \mid R_v \subseteq Y\}$. Hence, there is a child Y of $\mathit{parent}(X)$ such that $f(u) \geq \triangledown(Y)$. Then, we have $f(u) \geq \triangledown(X)$ or $f(u) \geq \triangledown(\mathit{sibling}(X))$. In the case where $f(u) \geq \triangledown(\mathit{sibling}(X))$, this also implies that $f(u) \geq \triangledown(X)$ because $\triangledown(X) \leq \triangledown(\mathit{sibling}(X))$. Therefore, $\xi(X)$, being equal to $f(u)$, is greater than or equal to $\triangledown(X)$.

3. If X is a dominant region for $(f, \prec)$, then $\xi(X) = \xi(\mathit{parent}(X))$ (second case of Definition 27). We will prove by induction that this lemma holds true for any dominant region for $(f, \prec)$. In the base step, we consider that $\mathit{parent}(X)$ is V. In the inductive step, we show that, if the property holds true for $\mathit{parent}(X)$, then it also holds true for X. Please note that, if $\mathit{parent}(X)$ is not a dominant region for $(f, \prec)$, the property holds for $\mathit{parent}(X)$ as proven in the previous case.

(a) Base step: if $\mathit{parent}(X)$ is V, then $\xi(X) = \xi(V) = \triangledown(V) + 1$ (first case of Definition 27). We can see that $\triangledown(V) \geq \triangledown(X)$ because, for any edge u such that $R_u \subseteq X$, we also have $R_u \subseteq V$. Then, $\xi(X)$, being equal to $\triangledown(V) + 1$, is greater than $\triangledown(X)$.

(b) Inductive step: let us assume that $\xi(\mathit{parent}(X)) \geq \triangledown(\mathit{parent}(X))$. Since $\xi(X) = \xi(\mathit{parent}(X))$, we have $\xi(X) \geq \triangledown(\mathit{parent}(X))$. We can affirm that, for any edge v in $E_{\prec}$ such that $R_v \subseteq X$, we also have $R_v \subseteq \mathit{parent}(X)$. Hence, $\triangledown(\mathit{parent}(X)) \geq \triangledown(X)$. Therefore, $\xi(X)$, being equal to $\xi(\mathit{parent}(X))$, is greater than or equal to $\triangledown(X)$.

□

Proof of Lemma 92. We will prove that, for any region X of $\mathcal{B}_\prec$, we have $\xi(X) = \vee\{\xi_f(M)$ such that M is a minimum of w included in $X\}$. Let X be a region of $\mathcal{B}_\prec$. We consider two cases: (1) there is a minimum of w included in X; and (2) there is no minimum of w included in X.

(1) If there is no minimum of w included in X, then X is not a dominant region for $(f, \prec)$. Then $\xi(X) = f(u)$ (third condition of Definition 27), where u is the building edge of $parent(X)$. The edge u is not a watershed-cut edge for $\prec$ because the child X of R_u does not include any minimum of w. Hence, since f is one-side increasing for $\prec$, by the statement 2 of Definition 18, we have $f(u) = 0$. Therefore, $\xi(X)$, being equal to $f(u)$, is also equal to $\vee\{\xi(M)$ such that M is a minimum of w included in $R\} = \vee\{\} = 0$.

(2) Let us assume that there is at least one minimum of w included in X. If X is a minimum of w, then $\xi(X) = \vee\{\xi_f(M)$ such that M is a minimum of w included in $X\} = \vee\{\xi_f(X)\}$.

In order to prove the case where X is not a minimum of w, we will first demonstrate that $\xi(X) \geq \vee\{\xi(Y) \mid Y \subseteq X\}$.

To prove that $\xi(X) \geq \vee\{\xi(Y) \mid Y \subseteq X\}$, it is enough to demonstrate that, for any region Z of $\mathcal{B}_\prec$, we have $\xi(Z) \geq \vee\{\xi(Y) \mid Y \textit{ is a child of } Z\}$. Let Z be a region of $\mathcal{B}_\prec$. If Z is a leaf region of $\mathcal{B}_\prec$, then $\xi(Z) \geq \vee\{\xi(Y) \mid Y \textit{ is a child of } Z\} = \vee\{\} = 0$ because, by Lemma 87, $\xi(Z)$ is in $\{0, \ldots, n\}$. Let us now assume that Z is not a leaf region of $\mathcal{B}_\prec$ and let Y be a child of Z. If Y is a dominant region for $(f, \prec)$, then $\xi(Y) = \xi(Z)$ and, consequently, $\xi(Z) \geq \xi(Y)$. Otherwise, if Y is not a dominant region for $(f, \prec)$, then $\xi(Y) = f(v)$, where v is the building edge of Z. By Lemma 93, $\xi(Z) \geq \triangledown(Z)$ and, consequently, $\xi(Z) \geq f(u)$. Hence, $\xi(Z) \geq \xi(Y)$.

We can now prove that $\xi(X) = \vee\{\xi_f(M)$ such that M is a minimum of w included in $X\}$ in the case where X is not a minimum of w. By Lemma 91, there is a minimum M of w such that $M \subset X$ and such that $\xi(M) = \xi(X)$. Let M be the minimum of w such that $\xi(M) = \xi(X)$. Since $\xi(X) \geq \vee\{\xi(Y) \mid Y \subseteq X\}$, we can say that $\xi(X) = \vee\{\xi_f(M')$ such that M' is a minimum of w included in $X\}$.

□

Lemma 94. *Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$ such that the approximated extinction map for $(f, \prec)$ is an extinction map for $\prec$. Then, f is one-side increasing for $\prec$.*

To prove Lemma 94, we will prove that the three conditions of Definition 18 for f to be one-side increasing for $\prec$ hold true. Those conditions are proven in properties 95, 96 and 97.

Property 95. *Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$. Let ξ be the approximated extinction map for $(f, \prec)$ such that ξ is an extinction map. Then $\{f(u) \mid u \in E_{\prec}\} = \{0, \ldots, n-1\}$.*

Proof. We need to prove that:

1. for any i in $\{0, \ldots, n-1\}$, there is an edge u in $E_{\prec}$ such that $f(u) = i$; and
2. for any edge u in $E_{\prec}$, we have $f(u)$ in $\{0, \ldots, n-1\}$.

Proof of 1*:*

For $i = 0$: Since ξ is an extinction map for $\prec$, by Property 23 (statement 3), for any leaf region R of $\mathcal{B}_{\prec}$, we have $\xi(R) = \vee\{\xi(M)$ such that M is a minimum of w included in $R\} = 0$. Let R be a leaf region. Then, by Definition 26, R is not a dominant region for $(f, \prec)$. Hence, $\xi(R) = f(u)$, where u is the building edge of the parent of R, and, since $\xi(R) = 0$, this implies that there exists an edge u in E such that $f(u) = 0$.

For i in $\{1, \ldots, n-1\}$: as ξ is an extinction map, by Property 23 (statement 1), we have $\{\xi(R) \mid R \textit{ is a region of } \mathcal{B}_{\prec}\} = \{0, \ldots, n\}$. Then, for any i in $\{1, \ldots, n-1\}$ there is a region R of $\mathcal{B}_{\prec}$ such that $\xi(R) = i$. Let i be any value in $\{1, \ldots, n-1\}$ and let R be a region of $\mathcal{B}_{\prec}$ such that $\xi(R) = i$. If R is not a dominant region for $(f, \prec)$, then $\xi(R) = f(u)$, where u is the building edge of the parent of R and, then, we can affirm that there exists an edge in $E_{\prec}$ whose weight for f is i. Otherwise, if R is a dominant region for $(f, \prec)$, then $\xi(R) = \xi(parent(R))$. If $parent(R)$ is not a dominant region for $(f, \prec)$, then $\xi(parent(R)) = \xi(v)$, where v is the building edge of the parent of $parent(R)$ and we have our property. Otherwise, if $parent(R)$ is a dominant region for $(f, \prec)$, then $\xi(parent(R)) = \xi(parent(parent(R)))$. We can see that, at some point, we will have $\xi(R) = \xi(parent \ldots (parent(R)))) = f(y)$ for an edge y in $E_{\prec}$. Therefore, there is an edge in $E_{\prec}$ whose weight for f is i.

Proof of 2: By contradiction, let us assume that there is an edge u in $E_{\prec}$ such that $f(u)$ is not in $\{0, \ldots, n-1\}$. We can affirm that any non leaf region of $\mathcal{B}_{\prec}$ has a child which is not a dominant region for $(f, \prec)$. So, we can affirm that there is a child X of R_u such that $\xi(X) = f(u)$. Since ξ is an extinction map, by Property 23 (statement 1), we have $\{\xi(R) \mid R \textit{ is a region of } \mathcal{B}_{\prec}\} = \{0, \ldots, n\}$. Then, $\xi(X) = f(u)$ should be in $\{0, \ldots, n\}$ as well. Therefore, the only value that $f(u)$ could take and that is not in $\{0, \ldots, n-1\}$ is n. So, let us assume that $f(u) = n$. In this case, the supremum descendant value of V for $(f, \prec)$ would be n and, consequently, $\xi(V)$ would be $n+1$, which contradicts the fact that $\{\xi(R) \mid R \textit{ is a region of } \mathcal{B}_\prec\} = \{0, \ldots, n\}$. Therefore, we may conclude that, for any edge u in $E_\prec$, we have $f(u)$ in $\{0, \ldots, n-1\}$.

□

Property 96. *Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$. Let ξ be the approximated extinction map for $(f, \prec)$ such that ξ is an extinction map. Then, for any edge u in $E_\prec$, the weight $f(u)$ is greater than zero if and only if u is a watershed-cut edge for $\prec$.*

Proof. Let u be an edge in $E_\prec$. If u is a watershed-cut edge for $\prec$, then there is at least one minimum of w included in each child of R_u. Hence, by Property 23, the value of each child of R_u in ξ is greater than zero. As there is at most one child of R_u that is a dominant region for $(f, \prec)$, then there is a child of R_u that is not a dominant region for $(f, \prec)$. Let X be a child of R_u that is not a dominant region for $(f, \prec)$. Then, by Definition 27, $\xi(X) = f(u)$. Therefore, since $\xi(X)$ is greater than zero, we have that $f(u)$ is greater than zero.

Now, let us assume that u is not a watershed-cut edge for $\prec$. Hence, there is a child X of R_u that do not include any minimum of w. Let X be a child of R_u that do not include any minimum of w. Then, by Definition 26, X is not a dominant region for $(f, \prec)$. Hence, by Definition 27, $\xi(X) = f(u)$. Since ξ is an extinction map and since there is no minimum of w included in X, we can affirm that $\xi(X) = 0$. Therefore, $f(u) = 0$. □

Property 97. *Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$. Let $\triangledown$ be a map from each region of $\mathcal{B}_\prec$ into its supremum descendant value (for $(f, \prec)$). Let ξ be the approximated extinction map for $(f, \prec)$ such that ξ is an extinction map. Then, for any u in $E_\prec$, there exists a child R of R_u such that $f(u)$ is greater or equal to the supremum descendant value $\triangledown(R)$.*

In order to prove Property 97, we first present properties 98 and 99.

Property 98. *Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$. Let ξ be the approximated extinction map for $(f, \prec)$ such that ξ is an extinction map. Then, for any region R of $\mathcal{B}_\prec$, $\xi(R) \geq \vee\{\xi(X) \mid X \subseteq R\}$.*

Proof. Direct result of the statement 3 of Property 23. □

Property 99. *Let $\prec$ be an altitude ordering for w and let f be a map from E into $\mathbb{R}$. Let ξ be the approximated extinction map for $(f, \prec)$ such that ξ is an extinction map for $\prec$. Then, for any edge u in $E_\prec$, we have $\xi(R_u) \geq f(u)$.*

Proof. Let u be an edge in $E_{\prec}$. There is a child of R_u that is not a dominant region for $(f, \prec)$. Let X be a child of R_u that is not a dominant region for $(f, \prec)$. Then, $\xi(X) = f(u)$. By Property 98, we have that $\xi(R_u) \geq \xi(X)$. Hence, $\xi(R_u) \geq f(u)$. □

Proof of Property 97. Let u be an edge in $E_{\prec}$. By Property 98, we have $\xi(R_u) \geq \{\xi(X) \mid X \subseteq R_u\}$. Then, by Property 99, we have $\xi(R_u) \geq \{f(v) \mid v$ is the building edge of a region $X \subseteq R\}$. At most one of the children of R_u is a dominant region for $(f, \prec)$. Let R be a child of R_u that is not a dominant region for $(f, \prec)$. As R is not a dominant region for $(f, \prec)$, then $\xi(R) = f(u)$. So, we will have $f(u) \geq \{f(v) \mid v$ is the building edge of $X \subseteq R\}$. Therefore, there exists a child R of R_u such that $f(u)$ is greater or equal to the supremum descendant value $\bigtriangledown(R)$.

□

Proof of Theorem 28. Let f be a map from E into $\mathbb{R}$, let $\prec$ be a lexicographic ordering for (w, f), and let ξ be the approximated extinction map for $(f, \prec)$. The map f is the saliency map of a hierarchical watershed of (G, w) if and only if the map ξ is an extinction map for $\prec$.

Forward implication: Let f be the saliency map of a hierarchical watershed of (G, w). Then, by Theorem 20, f is one side increasing for the lexicographic ordering $\prec$ for (w, f). Thus, by Lemma 85, ξ is an extinction map for $\prec$.

Backward implication: Let ξ be an extinction map for $\prec$. Then, by Property 94, f is one-side increasing for $\prec$. Hence, by Theorem 20, f is the saliency map of a hierarchical watershed of (G, w).

□

8.2.4 Proof of Property 30

(Property 30). *Let f be a map from E into $\mathbb{R}$, let $\prec$ be a lexicographic ordering for (w, f), and let $\mathcal{S}$ be the estimated sequence of minima for $(f, \prec)$. If f is the saliency map of a hierarchical watershed of (G, w), then f is the saliency map of a hierarchical watershed of (G, w) for $\mathcal{S}$.*

To prove Property 30, we first establish the following auxiliary property.

Property 100. *Let f be a map from E into $\mathbb{R}$ and let $\prec$ be an altitude ordering for w. Let ξ be the approximated extinction map for $(f, \prec)$, and let $\mathcal{S}$ be the estimated sequence of minima for $(f, \prec)$. If ξ is an extinction map for $\prec$, then ξ is the extinction map for $\mathcal{S}$ and $\prec$.*

Proof. Let ξ be an extinction map for $\prec$. Then, there exists a sequence $\mathcal{S}' = (M_1, \ldots, M_n)$ of minima of w such that ξ is the extinction map for $\mathcal{S}'$ and $\prec$. By the definition of extinction maps, for any i in $\{1, \ldots, n\}$, $\xi(M_i) = i$. Hence, for any i in $\{2, \ldots, n\}$, $\xi(M_i) > \xi(M_{i-1})$. Therefore, $\mathcal{S}'$ is a sequence of minima ordered according to their extinction values in ξ. Therefore, $\mathcal{S}'$ corresponds to the estimated sequence of minima for f and $\prec'$: $\mathcal{S}$. Thus, ξ is the extinction map for $\mathcal{S}$ and $\prec$. □

Proof of Property 30. Let f be the saliency map of a hierarchical watershed of (G, w). Let ξ be the approximated extinction map for $(f, \prec)$. Then, by Theorem 28, ξ is an extinction map. By Property 99, for any u in $E_\prec$, we have $\xi(R_u) \geq f(u)$. Hence, for any u in $E_\prec$, we have $f(u) = min\{\xi(R_u), f(u)\}$. By the definition of approximated extinction maps (Definition 27), we can conclude that, for any child X of $R(u)$, we have either $\xi(X) = \xi(R_u)$ or $\xi(X) = f(u)$. Hence, $f(u) = min\{\xi(R) \mid R \textit{ is a child of } R_u\}$. By Property 100, ξ is the extinction map for $\mathcal{S}$ and $\prec$. Hence, f maps any building edge u for $\prec$ into its persistence value for $(\mathcal{S}, \prec)$. Therefore, f is the saliency map of a hierarchical watershed for $\mathcal{S}$ (see discussion in Section 2.6.1). □

8.2.5 Proof of Theorem 33

(Theorem 33). *Let $\mathcal{H}$ be a hierarchy, let f be the saliency map of $\mathcal{H}$ and let $\prec$ be a lexicographic ordering for (w, f). The following statements hold true:*

1. *The hierarchy $\mathcal{H}$ is a hierarchical watershed of (G, w) if and only if the watersheding $\omega(f)$ of f (for $\prec$) is equal to f.*
2. *The watersheding $\omega(f)$ of f is the saliency map of a hierarchical watershed of (G, w).*
3. *The watersheding $\omega(\omega(f))$ of $\omega(f)$ is equal to $\omega(f)$.*

To prove Theorem 33, we first present the following lemma.

Let (G, w) be a tree, let $\mathcal{S}$ be a sequence of minima of w, and let f be the saliency map of a hierarchical watershed of (G, w) for $\mathcal{S}$. Let $\prec$ be a lexicographic ordering for (w, f). Then f is the saliency map of the hierarchy induced by $(\mathcal{S}, \prec)$.

Proof of Theorem 33. Statement 1:

Forward implication: let $\mathcal{H}$ be a hierarchical watershed of (G, w) and let $\mathcal{S}$ be the estimated sequence of minima for $(f, \prec)$. By the definition of the watersheding operator, for any edge u in E, the value $\omega(f)(u)$ is the persistence value of u for $(\mathcal{S}, \prec)$. Hence, as discussed in Section 2.6.1, the map $\omega(f)$ is the saliency map of the hierarchy induced by $(\mathcal{S}, \prec)$. We will prove that f is also the saliency map of the hierarchy induced by $(\mathcal{S}, \prec)$. By Property 30, the map f is the saliency map of a hierarchical watershed of (G, w) for $\mathcal{S}$. Therefore, by Lemma 8.2.5, the map f is the saliency map of the hierarchy induced by $(\mathcal{S}, \prec)$. Thus, the watersheding $\omega(f)$ of f is equal to f.

Backward implication: Let $\omega(f)$ be equal to f. Let $\mathcal{S}$ be the estimated sequence of minima for $(f, \prec)$. By Theorem 32, the map $\omega(f)$ is the saliency map of a hierarchical watershed of (G, w) for $\mathcal{S}$. Hence, since $\omega(f)$ is equal to f, the map f is the saliency map of a hierarchical watershed of (G, w) for $\mathcal{S}$. Then, the hierarchy $\mathcal{H}$ is a hierarchical watershed of (G, w).

Statement 2: Let $\mathcal{S}$ be the estimated sequence of minima for f and $\prec$. By Theorem 32, the watersheding $\omega(f)$ of f is the saliency map of a hierarchical watershed of (G, w) for $\mathcal{S}$. Hence, $\omega(f)$ is the saliency map of a hierarchical watershed of (G, w).

Statement 3: By the statement 2, the watersheding $\omega(f)$ of f is the saliency map of a hierarchical watershed of (G, w). Hence, by the first statement, the watersheding $\omega(\omega(f))$ of $(\omega(f)$ is equal to $(\omega(f)$.

□

8.3 Proofs of theorem and properties of Chapter 5

8.3.1 Proof of Property 37

(**Property 37).** *Let $\mathcal{H}$ be a hierarchical watershed of (G, w) and let m be the number of maximal regions of $\mathcal{B}_\prec$ for the saliency map of $\mathcal{H}$. The probability of $\mathcal{H}$ knowing w is:*

$$p(\mathcal{H} \mid w) = \frac{2^m}{|\mathcal{M}_w|}. \tag{8.2}$$

In order to prove Property 37, we establish the following auxiliary lemmas.

Lemma 101. *Let (G, w) be a weighted graph and let $\prec$ be an altitude ordering for w. Let u be a watershed-cut for $\prec$ and let ℓ be the number of minima included in R_u. There are $\ell - 1$ watershed-cut edges v for $\prec$ such that $R_v \subseteq R_v$.*

Lemma 102. *Let (G, w) be a tree with a unique altitude ordering and let $\prec$ be the altitude ordering for w. Let $\mathcal{S}$ be a sequence of minima of w and let ϵ be the extinction map for $(\mathcal{S}, \prec)$. Let f be the saliency map of the hierarchical watershed of (G, w) for $\mathcal{S}$. For any edge u of G, the value $f(u)$ is $min\{\epsilon(X), \epsilon(Y)\}$, where X and Y are the children of R_u.*

Proof. By Property 6, since f is the saliency map of the hierarchical watershed of (G, w) for $\mathcal{S}$, then f is the saliency map of the hierarchy induced by $\mathcal{S}$ and by an altitude ordering for w. Since $\prec$ is the unique altitude ordering for w, then f is the saliency map of the hierarchy induced by $\mathcal{S}$ and $\prec$. Hence, $f(u)$ is the persistence value of u for $(\mathcal{S}, \prec)$. Therefore, the value $f(u)$ is $min\{\epsilon(X), \epsilon(Y)\}$, where X and Y are the children of R_u. □

Lemma 103. *Let (G, w) be a tree with a unique altitude ordering and let $\prec$ be the altitude ordering for w. Let $\mathcal{S}$ be a sequence of minima of w and let ϵ be the extinction map for $(\mathcal{S}, \prec)$. Let f be the saliency map of the hierarchical watershed of (G, w) for $\mathcal{S}$. Let u be a watershed-cut edge for $\prec$. We have $f(u) = \epsilon(M)$, where M is a minimum of w included in R_u.*

Proof. Direct implication of Lemma 102. □

Lemma 104. *Let (G, w) be a tree with a unique altitude ordering and let $\prec$ be the altitude ordering for w. Let $\mathcal{S}$ be a sequence of minima of w and let ϵ be the extinction map for $(\mathcal{S}, \prec)$. Let f be the saliency map of the hierarchical watershed of (G, w) for $\mathcal{S}$. Let u be a watershed-cut edge for $\prec$ and let M be a minimum of w. If M is not the minimum of greatest extinction value among the minima included in R_u, there is a watershed-cut edge v such that $R_v \subseteq R_u$ and such that $f(v) = \epsilon(M)$.*

Proof. Let ℓ be the number of minima included in R_u. By Lemma 101, there are $\ell - 1$ watershed-cut edges v such that R_v is included in R_u. By Lemma 103, the weight of each watershed-cut edge v such that $R_v \subseteq R_u$ is the extinction value of a minimum included in R_u. Moreover, by Lemma 19, f is one-side increasing for $\prec$ and, consequently, the watershed-cut edges have pairwise distinct edge weights in f. Therefore, for $\ell - 1$ minima included in R_u, there is a watershed-cut edge v such that $R_v \subseteq R_u$ and such that $f(u)$ is the extinction value of this minimum. Hence, there is only one minimum M included in R_u such that there is no watershed-cut edge v such that $R_v \subseteq R_u$ and $f(u) = \epsilon(M)$. Let v be a watershed-cut edge such that $R_v \subseteq R_u$. Let M' be the only minimum included in R_v such that there is no watershed-cut edge v such that $R_v \subseteq R_u$ and $f(u) = \epsilon(M')$. By Lemma 102, the value $f(v)$ is $min\{\epsilon(X), \epsilon(Y)\}$, where X and Y are the children of R_v. Hence, we can conclude that the minimum M' is the minimum of maximal extinction value among the minima include in R_u. Therefore, if M is not the minimum of greatest extinction value among the minima included in R_u, there is a watershed-cut edge v such that $R_v \subseteq R_u$ and such that $f(v) = \epsilon(M)$. □

Lemma 105. *Let (G, w) be a tree with a unique altitude ordering and let $\prec$ be the altitude ordering for w. Let $\mathcal{S}$ be a sequence of minima of w and let ϵ be the extinction map for $(\mathcal{S}, \prec)$. Let f be the saliency map of the hierarchical watershed of (G, w) for $\mathcal{S}$. Let u be a watershed-cut edge for $\prec$. Let M be the minimum of maximum extinction value among the minima included in R_u. Then, for any watershed-cut edge v for $\prec$ such that $R_v \subseteq R_u$, we have $f(v) < \epsilon(M)$.*

Proof. Let v be a watershed-cut edge v such that $R_v \subseteq R_u$. By Lemma 103, the value $f(v)$ is the extinction value of a minimum included in R_u. By Lemma 104, $f(u)$ is different from the extinction value of M. As the minima of w have pairwise distinct extinction values and since the extinction value of M is maximal among the minima included in R_u, we conclude that $f(v) < \epsilon(M)$. □

Lemma 106. *Let (G, w) be a weighted graph and let $\prec$ be the unique altitude ordering for w. Let $\mathcal{S}$ be a sequence of minima of w and let ϵ be the extinction map for $(\mathcal{S}, \prec)$. Let f be the saliency map of the hierarchical watershed of (G, w) for $\mathcal{S}$. Let u be a watershed-cut edge for $\prec$. Let X and Y be the children of R_u such that $\epsilon(X) > \epsilon(Y)$. Let M_x be the minimum included in X such that $\epsilon(X) = \epsilon(M_x)$ and let M_y be the minimum included in Y such that $\epsilon(Y) = \epsilon(Y_x)$. The region R_u is a maximal region of $\mathcal{B}_\prec$ for f if and only if:*

1. *either M_x is the only minimum included in X; or*
2. *$\epsilon(M'_x) < \epsilon(M_y)$, where M'_x is the minimum with the second greatest extinction value among all minima included in X.*

Proof. We first prove the forward implication. We consider the conditions 1 and 2 separately.

1. Let us assume that M_x is the only minimum included in X. To prove that u is a maximal region of $\mathcal{B}_\prec$ for f, we will first prove that $f(u) > \{f(v) \mid R_v \subset X\}$ and, then, we will prove that $f(u) > \{f(v) \mid R_v \subset Y\}$. Let r be the building edge of X. By Definition 2, r is not a watershed-cut edge for $\prec$. Hence, since f is one-side increasing for $\prec$ by Lemma 19, we can say that $f(r) = 0$ by the second statement of Definition 18. Therefore, by the third statement of Definition 18, for any edge r' such that $R_{r'} \subset R_v$, we have $f(r') = 0$. Since u is a watershed-cut edge for $\prec$, we have $f(u) > 0$ by the second statement of Definition 18. We now prove that $f(u) > \{f(v) \mid R_v \subset Y\}$. By Lemma 102, we have $f(u) = min\{\epsilon(M_x), \epsilon(M_y)\}$. By

our assumption that $\epsilon(M_x) > \epsilon(M_y)$, we have $f(u) = \epsilon(M_y)$. Hence, as $\epsilon(M_y)$ is the greatest extinction value among the extinction values of the minima included in Y, Lemma 105, we can say that $f(u) > \{f(v) \mid R_v \subset Y\}$. Therefore, $f(u) > \{f(v) \mid R_v \subset R_u\}$ and R_u is a maximal region for $\mathcal{B}_\prec$.

2. Let us now assume that $\epsilon(M'_x) < \epsilon(M_y)$ where M'_x is the minimum with the second greatest extinction value among all minima included in X. To prove that u is a maximal region of $\mathcal{B}_\prec$ for f, we will first prove that $f(u) > \{f(v) \mid R_v \subset X\}$ and, then, we will prove that $f(u) > \{f(v) \mid R_v \subset Y\}$. As affirmed previously, we have $f(u) = \epsilon(M_y)$. By Lemma 105, we may say that there is no edge v such that $R_v \subseteq X$ and such that $f(v) = \epsilon(X)$. Hence, for any edge v such that $R_v \subseteq X$, we have $f(v) < \epsilon(X)$, which implies that $f(v) \leq \epsilon(M'_x)$ by Lemma 103. Then, by our assumption that $\epsilon(M_x) > \epsilon(M_y)$, for any edge v such that $R_v \subseteq X$, we have $f(v) \leq \epsilon(M_y)$. Since $f(u) = \epsilon(M_y)$, we have $f(u) > \{f(v) \mid R_v \subset X\}$. We now prove that $f(u) > \{f(v) \mid R_v \subset Y\}$. By Lemma 102, we have $f(u) = min\{\epsilon(M_x), \epsilon(M_y)\}$. By our assumption that $\epsilon(M_x) > \epsilon(M_y)$, we have $f(u) = \epsilon(M_y)$. Hence, as $\epsilon(M_y)$ is the greatest extinction value among the extinction values of the minima included in Y, by Lemma 105, we can say that $f(u) > \{f(v) \mid R_v \subset Y\}$. Therefore, $f(u) > \{f(v) \mid R_v \subset R_u\}$ and R_u is a maximal region for $\mathcal{B}_\prec$.

We now prove the backward implication. Let R_u be a maximal region. Then $f(u) > \{f(v) \mid R_v \subset R_u\}$. By contradiction, let us assume that M_x is not the only minimum included in X and that $\epsilon(M'_x) \geq \epsilon(M_y)$, where M'_x is the minimum with the second greatest extinction value among all minima included in X. In this case, by Lemma 104, there is an edge v' such that $R_{v'} \subseteq X$ and such that $f(v') = \epsilon(M'_x)$. By Lemma 102, we have $f(u) = min\{\epsilon(M_x), \epsilon(M_y)\}$. By our assumption that $\epsilon(M_x) > \epsilon(M_y)$, we have $f(u) = \epsilon(M_y)$. Hence, there is an edge v' such that $R_{v'} \subset R_u$ and such that $f(v') \geq f(u)$, which contradicts our assumption that R_u is a maximal region. Therefore, either M_x is the only minimum included in X or $\epsilon(M'_x) < \epsilon(M_y)$, where M'_x is the minimum with the second greatest extinction value among all minima included in X. □

Definition 107 (least common ancestor). *Let $\mathcal{H}$ be a hierarchy on V. Let X and Y be two distinct regions of $\mathcal{H}$. The* least common ancestor (LCA) of X and Y *is the region R of $\mathcal{H}$ such that $X \subset R$ and $Y \subset R$.*

Lemma 108. *Let (G, w) be a weighted graph and let $\prec$ be the unique altitude ordering for w. Let $\mathcal{S}$ be a sequence of minima of w. Let ϵ be the extinction map for $(\mathcal{S}, \prec)$. Let M_x and M_y be two minima of w. Let u be the building edge of the LCA of M_x and*

M_y. Let X and Y be the children of R_u such that M_x and M_y are included in X and Y, respectively. Let $\mathcal{S}'$ be the sequence of minima resulting from swapping the positions of M_x and M_y in $\mathcal{S}$. The saliency map of the hierarchical watersheds for $\mathcal{S}$ and $\mathcal{S}'$ are equal if and only the two following statements hold true:

1. *R_u is a maximal region of $\mathcal{B}_\prec$ for the saliency map of $\mathcal{H}$; and*
2. *M_x (resp. M_y) is the minimum of w of greatest extinction value among the minima included in X (resp. Y).*

Proof. We will first prove the forward implication. Let ϵ' be the extinction map for $(\mathcal{S}', \prec)$. Let u be the building edge of a maximal region of $\mathcal{B}_\prec$ for the saliency map of $\mathcal{H}$ and let X and Y be the children of R_u. Let M_x (resp. M_y) be the minimum of w of greatest extinction value among the minima included in X (resp. Y). Without loss of generality, let us assume that $\epsilon(M_x) > \epsilon(M_y)$. Let f and f' be the saliency maps of the hierarchical watersheds for $\mathcal{S}$ and $\mathcal{S}'$, respectively. We will prove that f and f' are equal: for any edge v, we will prove that $f(v) = f'(v)$. Let v be an edge in E. We will consider the following cases: (1) $R_v \cap R_u = \emptyset$; (2) $R_v = R_u$; (3) $R_u \subset R_v$; and (4) $R_v \subset R_u$.

1. $R_v \cap R_u = \emptyset$. In this case, for any region Z such that $Z \subseteq R_v$, we have $\epsilon'(Z) = \epsilon(Z)$ because the position of the minima included in R_u are the same in the sequences $\mathcal{S}$ and $\mathcal{S}'$. Since $f'(v)$ is defined by the extinction value of the children of R_v, we may affirm that $f(u) = f(u')$.

2. $R_v = R_u$. By Lemma 102, the value $f(u)$ is $min\{\epsilon(X), \epsilon(Y)\}$. Let M'_x be the minimum of second greatest extinction value among the minima included in X. By Property 106, since $\epsilon(M_x) > \epsilon(M_y)$, we have $\epsilon(M'_x) < \epsilon(M_y)$. Therefore, we can say that M_x (resp. M_y) is still the minimum of w of greatest extinction value for ϵ' among the minima included in X (resp. Y). Hence, we have that $\epsilon'(X) = \epsilon'(M_x) = \epsilon(Y)$ and that $\epsilon'(Y) = \epsilon'(M_y) = \epsilon(X)$. Hence, $f'(u) = min\{\epsilon'(X), \epsilon'(Y)\} = min\{\epsilon(Y), \epsilon(X)\} = f(u)$.

3. $R_u \subset R_v$. The only minima that had their extinction values changed in ϵ' with respect to ϵ were two minima M_x and M_y included in R_u. Hence, the greatest extinction value among the minima included in R_u is still the same even though the minimum carrying this extinction value has changed. Therefore, the extinction value of both children of R_v has not been changed and $f(u) = f(u')$.

4. $R_v \subset R_u$. We will consider the two following cases: (a) $R_v \subseteq X$; and (b) $R_v \subseteq Y$.

(a) $R_v \subseteq X$. If M_x is not included in R_v, we can conclude that the extinction value for ϵ' of all regions included in R_v have not been changed with respect to ϵ, which implies that $f'(u) = f(u)$. Otherwise, let us assume that M_x is included in R_v. If $M_x = R_v$, then v is not a watershed-cut edge for $\prec$ and we have $f(u) = f'(u) = 0$ by the second statement of Definition 18. Otherwise, let R' be the child of R_v that includes M_x. By our assumption, M_x is the minimum of w of greatest extinction value among the minima included in X. Therefore, $\epsilon(R') = \epsilon(M_x)$. Moreover, $f(v)$, being $min\{\epsilon(R'), \epsilon(sibling(R'))\}$, is equal to $\epsilon(sibling(R'))$. Let M'_x be the minimum of w of second greatest extinction value among the minima included in X. By Property 106, as R_u is a maximal region and as $\epsilon(M_x) > \epsilon(M_y)$, we have that $\epsilon(M'_x) < \epsilon(M_y)$. Therefore, we may say that $\epsilon(sibling(R')) < \epsilon(M_y)$. Consequently, $\epsilon'(R')$, being equal to $\epsilon(M_y)$, is greater than $\epsilon(sibling(R'))$. Therefore, $f'(v) = min\{\epsilon'(R'), \epsilon(sibling(R'))\} = \epsilon(sibling(R')) = f(v)$.

(b) $R_v \subseteq Y$. If M_y is not included in R_v, we can conclude that the extinction value for ϵ' of all regions included in R_v have not been changed with respect to ϵ, which implies that $f'(u) = f(u)$. Otherwise, let us assume that M_y is included in R_v. Hence, there is a child R' of R_v that includes M_y and whose extinction value is $\epsilon(M_y)$. By our assumption, M_y is the minimum of w of greatest extinction value among the minima included in Y. Hence, $f(v)$, being $min\{\epsilon(R'), \epsilon(sibling(R'))\}$, is equal to $\epsilon(sibling(R'))$. By our hypothesis, we have that $\epsilon'(M_x) = \epsilon(M_y) = \epsilon(Y)$ and $\epsilon'(M_y) = \epsilon(M_x) = \epsilon(X)$. Since $\epsilon(X) > \epsilon(Y)$ by our hypothesis, we have $\epsilon'(R') > \epsilon(R')$. Therefore, $\epsilon'(R')$, alike $\epsilon(R')$, is also greater than $\epsilon(sibling(R'))$. Hence, $f'(v) = min\{\epsilon'(R'), \epsilon(sibling(R'))\} = \epsilon(sibling(R')) = f(v)$.

We now prove the backward implication. Let f and f' be equal. Using Lemma 106, we will prove by contradiction that R_u is a maximal region of $\mathcal{B}_\prec$ for f and that M_x (resp. M_y) is the minimum of w of greatest extinction value among the minima included in X (resp. Y). Without loss of generality, let us assume that $\epsilon(M_x) > \epsilon(M_y)$.

By contradiction, let us first assume that M_x is not the minimum of greatest extinction value among the minima included in X. Then, by Lemma 104, there is an edge v such that $R_v \subseteq X$, $M_x \subset R_v$ and $f(v) = \epsilon(M_x)$. We know that $\epsilon'(M_x) = \epsilon(Y)$, which is greater than $\epsilon(M_x)$. Hence, there is no minimum included in R_v whose extinction value for $\mathcal{S}'$ is equal to $\epsilon(M_x)$. Therefore, by Lemma 105, $f'(v) < \epsilon(M_x)$ and, consequently, $f'(v) \neq f(v)$, which contradicts our assumption. Hence, M_x is the minimum of greatest extinction value among the minima included in X. The same argument can be used to prove that M_y is the minimum of greatest extinction value among the minima included in Y.

Now, by contradiction, let us assume that R_u is not a maximal region of $\mathcal{B}_\prec$ for f. If M_x (resp. M_y) is not the minimum of greatest extinction value among the minima included in X (resp. Y), we have that f and f' are not equal, as shown in the previous paragraph. Hence, let M_x (resp. M_y) be the minimum of greatest extinction value among the minima included in X (resp. Y). Let M'_x be the minimum included in X with the second greatest extinction value among all minima included in X. As R_u is not a maximal region, by Lemma 106, we have that $\epsilon(M'_x) \geq \epsilon(Y)$. We know that $\epsilon'(M_x) = \epsilon(M_y) = \epsilon(Y)$ and that $\epsilon'(M_y) = \epsilon(M_x) = \epsilon(X)$. Since $\epsilon(M'_x) \geq \epsilon(Y)$ and since $\epsilon'(M_x) = \epsilon(Y)$, then M'_x became the minimum of greatest extinction value for $\mathcal{S}'$ among the minima included in X. However, $\epsilon'(M'_x)$ is still less than $\epsilon'(M_y) = \epsilon(M_x)$. In the end, we will have $\epsilon'(X) = \epsilon(M'_x)$ and $\epsilon'(Y) = \epsilon(M_x)$. Since $f'(u) = min\{\epsilon'(X), \epsilon'(Y)\}$, we have that $f'(u) = \epsilon(M'_x)$. By our assumption that $\epsilon(M_x) > \epsilon(M_y)$, we have that $f(u) = \epsilon(M_y)$. Hence, $f'(u) \neq f(u)$, which contradicts our assumption that f and f' are equal. Therefore, R_u is a maximal region of $\mathcal{B}_\prec$ for f. □

Proof of Property 37. Let f be the saliency map of $\mathcal{H}$. By Property 35, $p(\mathcal{H} \mid w)$ is equal to $\frac{|\mathcal{S}_\mathcal{H}|}{|\mathcal{M}_w|}$. Hence, we need to prove that $|\mathcal{S}_\mathcal{H}|$ is equal to 2^m. To this end, we will prove that, given any sequence $\mathcal{S}$ in $\mathcal{S}_w(\mathcal{H})$, we can obtain another sequence in $\mathcal{S}_\mathcal{H}$ only by, for each maximal region R of $\mathcal{B}_\prec$ for f, swapping the order of two minima included in R. By Lemma 108, for each maximal region R of $\mathcal{B}_\prec$ for f, there is only one pair of minima included in R that can be swapped in the sequence $\mathcal{S}$ without changing the resulting saliency map. Moreover, the swapping of two minima is possible only if their LCA R' is a maximal region of $\mathcal{B}_\prec$ and if they are the minima of maximal extinction value included in each child of the region R'. Therefore, each maximal region doubles the number of permutations of $\mathcal{S}$ that result in a sequence in $\mathcal{S}_w(\mathcal{H})$. Thus, there are 2^m sequences of minima in $\mathcal{S}_w(\mathcal{H})$. □

8.4 Proofs of theorem and properties of Chapter 7

8.4.1 Proof of Property 46

(Property 46). *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two hierarchical watersheds of (G, w) and let $\prec$ be an altitude ordering for (G, w) such that both $\Phi(\mathcal{H}_1)$ and $\Phi(\mathcal{H}_2)$ are one-side increasing for $\prec$. Then the hierarchy $\mathcal{H}_\curlywedge = \mathcal{QFZ}(G, \curlywedge(\Phi(\mathcal{H}_1), \Phi(\mathcal{H}_2)))$ is a flattened hierarchical watershed of (G, w).*

Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two hierarchical watersheds of (G, w) and let $\prec$ be an altitude ordering for w such that both $\Phi(\mathcal{H}_1)$ and $\Phi(\mathcal{H}_2)$ are one-side increasing for $\prec$. Let f_3 denote the map $\curlywedge(\Phi(\mathcal{H}_1), \Phi(\mathcal{H}_2))$. We will prove that the hierarchy $\mathcal{QFZ}(G, f_3)$ is a flattened hierarchical watershed of (G, w). To this end, by Property 22, we will prove that the following statements hold true:

1. $(V, E_\prec)$ is a MST of (G, f_3), where $E_\prec$ is the set of building edges for $\prec$; and
2. for any edge u in $E_\prec$, if u is not a watershed-cut edge for $\prec$, then $f_3(u) = 0$; and
3. for any edge u in $E_\prec$, there exists a child R of R_u such that $f_3(u) \geq \vee\{f_3(v)$ such that R_v is included in $R\}$, where $\vee\{\} = 0$.

The following Lemmas 109, 112 and 113 prove respectively that the conditions 1, 2 and 3 for $\mathcal{QFZ}(G, f_3)$ to be a flattened hierarchical watershed of (G, w) hold true.

Lemma 109. *Let f_1 and f_2 be two maps from E into $\mathbb{R}$ and let G' be a subgraph of G such that G' is a MST of both (G, f_1) and (G, f_2). Then G' is also a MST of $(G, \curlywedge(f_1, f_2))$.*

In order to prove Lemma 109, we define cycles in the context of graphs and we state two well-known properties of spanning trees in Lemmas 110 and 111.

Let x and y be two vertices in V and let $\pi = (x_0, \ldots, x_p)$ be a path from x to y. For any edge $u = \{x_{i-1}, x_i\}$ for i in $\{1, \ldots, p\}$, we say that u *is in* π or that π *includes* u. We say that π *is a cycle* if $x_0 = x_p$ and $p > 1$.

Lemma 110. *Let G' be a spanning tree of (G, w) and let u be an edge in $E \setminus E(G')$. Then $(V, E(G') \cup \{u\})$ contains a cycle π that includes u.*

Lemma 111. *Let G' be a spanning tree of a weighted graph (G, f). Let u be an edge in $E \setminus E(G')$ and let π be the cycle of $(V, E(G') \cup \{u\})$ which includes u. The graph G' is a MST of (G, f) if and only if $f(u) \geq f(v)$ for any edge v in π.*

Proof of Lemma 109. Let f_3 denote the map $\curlywedge(f_1, f_2)$. Let u be an edge in $E \setminus E(G')$. As G' is a spanning tree, by Lemma 110, the graph $(V, E(G') \cup \{u\})$ contains a cycle π which includes the edge u. Since G' is a MST for (G, f_1) and for (G, f_2), by the forward implication of Lemma 111, for any edge v in the cycle π, we have $f_1(v) \leq f_1(u)$ and $f_2(v) \leq f_2(u)$. Therefore, for any edge v in the cycle π, we have $min(f_1(v), f_2(v)) \leq min(f_1(u), f_2(u))$ and, consequently, $f_3(v) \leq f_3(u)$. Hence, for any edge v in π, we have $f_3(u) \geq f_3(v)$. Thus, by the backward implication of Lemma 111, G' is a MST of (G, f_3). □

The following lemma proves that the condition 2 for $\mathcal{QFZ}(G, f_3)$ to be a flattened hierarchical watershed hold true.

Lemma 112. *Let f_1 and f_2 be two maps from E into $\mathbb{R}$ and let $\mathcal{B}$ be a binary partition hierarchy of (G, w) such that f_1 and f_2 are one-side increasing for $\prec$. Let f_3 denote the map $\curlywedge(f_1, f_2)$. Then for any edge u in $E_\prec$, if u is not a watershed-cut edge for $\prec$, then $f_3(u) = 0$.*

Proof. Let u be an edge in $E_\prec$. If u is not a watershed-cut edge for $\prec$, then, by the statement 2 of Definition 18, we have $f_1(u) = 0$ and $f_2(u) = 0$. Therefore, $f_3(u) = min(0, 0) = 0$. □

The following lemma proves that the condition 3 for $\mathcal{QFZ}(G, f_3)$ to be a flattened hierarchical watershed holds true.

Lemma 113. *Let f_1 and f_2 be two maps from E into $\mathbb{R}$ and let $\mathcal{B}$ be a binary partition hierarchy of (G, w) such that f_1 and f_2 are one-side increasing for $\prec$. Let f_3 denote $\curlywedge(f_1, f_2)$. Then, for any building edge u of $\mathcal{B}$, there exists a child R of R_u such that $f_3(u) \geq \vee\{f_3(v)$ such that $R_v \subseteq R\}$.*

Proof. Since f_1 (resp. f_2) is one-side increasing for $\prec$, by the statement 3 of Definition 18, we have that, for any building edge u of $\mathcal{B}$, $f_1(u) \geq \vee\{f_1(v) \mid R_v \subseteq X\}$ (resp. $f_2(u) \geq \vee\{f_2(v) \mid R_v \subseteq X\}$) for a child X of R_u. We need to prove that, for any building edge u of $\mathcal{B}$, $f_3(u) \geq \vee\{f_3(v) \mid R_v \subseteq X\}$ for a child X of R_u. Let u be a building edge of $\mathcal{B}$. As $f_3(u) = min(f_1(u), f_2(u))$, we should consider the following cases: (1) $f_3(u) = f_1(u)$; and (2) $f_3(u) = f_2(u)$.

1. Let us assume that $f_3(u) = f_1(u)$. Let X and Y be the children of R_u. If $f_1(u) \geq \vee\{f_1(v) \mid R_v \subseteq X\}$ (resp. $f_1(u) \geq \vee\{f_1(v) \mid R_v \subseteq Y\}$), we can affirm that $f_3(u) \geq \vee\{f_1(v) \mid R_v \subseteq X\}$ (resp. $f_3(u) \geq \vee\{f_1(v) \mid R_v \subseteq Y\}$) as well. Since $f_3(e) = min(f_1(e), f_2(e))$ for any edge e in E, we can affirm that $f_3(e) \leq f_1(e)$ for any edge e in E and, therefore, $f_3(u) \geq \vee\{f_3(v) \mid R_v \subseteq X\}$ (resp. $f_3(u) \geq \vee\{f_3(v) \mid R_v \subseteq Y\}$). Therefore, this condition holds true for the child X (resp. Y) of R_u.

2. Let us assume that $f_3(u) = f_2(u)$. The same reasoning of (1) can be applied in this case.

We can conclude that, for any building edge u of $\mathcal{B}$, we have $f_3(u) \geq \vee\{f_3(v) \mid R_v \subseteq R\}$ for a child R of R_u. □

Proof of Property 46. By Lemma 69, we can affirm that $(V, E_\prec)$ is a MST of both $(G, \Phi(\mathcal{H}_1))$ and $(G, \Phi(\mathcal{H}_2))$. Let f_3 denote the map $\curlywedge(\Phi(\mathcal{H}_1), \Phi(\mathcal{H}_2))$. By Lemma 109, $(V, E_\prec)$ is a MST of (G, f_3) as well, which proves the first condition for $\mathcal{QFZ}(G, f_3)$

to be a flattened hierarchical watershed of (G, w). By Lemmas 112 and 113, the second and third conditions for $\mathcal{QFZ}(G, f_3)$ to be a flattened hierarchical watershed of (G, w) hold true. Therefore, $\mathcal{QFZ}(G, f_3)$ is a flattened hierarchical watershed of (G, w). $\square$

8.4.2 Proof of Property 47

(**Property 47).** *Let C be a combining function, let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two hierarchical watersheds of (G, w) and let $\prec$ be and altitude ordering for (G, w) such that both $\Phi(\mathcal{H}_1)$ and $\Phi(\mathcal{H}_2)$ are one-side increasing for $\prec$. The combination of $\mathcal{H}_1$ and $\mathcal{H}_2$ with C is a flattened hierarchical watershed of (G, w) if $C(0, 0) = 0$ and if, for any a, b, c, d in $\{0, \ldots, n-1\}$, we have:*

1. *$C(a, b) = C(b, a)$; and*

2. *if $min(a, b) < min(c, d)$, then $C(a, b) < C(c, d)$; and*

3. *if $min(a, b) = min(c, d)$ and $max(a, b) < max(c, d)$, then $C(a, b) \leq C(c, d)$.*

Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two hierarchical watersheds of (G, w) and let $\prec$ be an altitude ordering for w such that both $\Phi(\mathcal{H}_1)$ and $\Phi(\mathcal{H}_2)$ are one-side increasing for $\prec$. Let C be a positive function from $\mathbb{R}^2$ into $\mathbb{R}$ such that, for any a, b, c and d in $\{0, \ldots, n-1\}$, we have:

1. $C(0, 0) = 0$; and

2. $C(a, b) = C(b, a)$; and

3. if $min(a, b) = min(c, d)$ and $max(a, b) < max(c, d)$ then $C(a, b) \leq C(c, d)$; and

4. if $min(a, b) < min(c, d)$ then $C(a, b) < C(c, d)$.

Let f_3 denote the map $C(\Phi(\mathcal{H}_1), \Phi(\mathcal{H}_2))$. We want to prove that the hierarchy $\mathcal{QFZ}(G, f_3)$ is a flattened hierarchical watershed of (G, w). By Property 22, we need to prove that there exists a binary partition hierarchy $\mathcal{B}'$ of (G, w) such that the following statements hold true:

1. $(V, E(\mathcal{B}'))$ is a MST of (G, f_3); and

2. for any edge u in $E(\mathcal{B}')$, if u is not a watershed-cut edge for $\mathcal{B}'$, then $f_3(u) = 0$; and

3. for any edge u in $E(\mathcal{B}')$, there exists a child R of R_u such that $f_3(u) \geq \vee\{f_3(v)$ such that R_v is included in $R\}$, where $\vee\{\} = 0$.

The proof of this property follows the same idea of the proof of Property 46. To prove Property 47, we establish the following auxiliary lemma.

Lemma 114. *Let C be a function from $\mathbb{R}^2$ into $\mathbb{R}$ such that, for any two real values x and y, we have $C(x, y) = C(y, x)$. Let a, b, c and d be four real values. If $min(a, b) = min(c, d)$ and $max(a, b) = max(c, d)$, then $C(a, b) = C(c, d)$.*

Proof. As $min(a, b) = min(c, d)$ and $max(a, b) = max(c, d)$, then either we have (i) $a = c$ and $b = d$ which implies that $C(a, b) = C(c, d)$; or (ii) $c = b$ and $d = a$ which implies that $C(c, d) = C(b, a)$, which, by our hypothesis on C, is equal to $C(a, b)$. Hence, we have $C(a, b) = C(c, d)$. □

The following three lemmas prove that the conditions 1, 2 and 3 for $\mathcal{QFZ}(G, f_3)$ to be a flattened hierarchical watershed of (G, w) hold true.

Lemma 115. *Let C be a positive function such that, for any a, b, c and d in $\{0, \dots, n-1\}$, we have:*

1. *$C(0, 0) = 0$; and*

2. *$C(a, b) = C(b, a)$; and*

3. *if $min(a, b) = min(c, d)$ and $max(a, b) < max(c, d)$ then $C(a, b) \leq C(c, d)$; and*

4. *if $min(a, b) < min(c, d)$ then $C(a, b) < C(c, d)$.*

Let f_1 and f_2 be the saliency maps of two hierarchies on V and let G' be a subgraph of G such that G' is a MST of both (G, f_1) and (G, f_2). Then G' is also a MST of $(G, C(f_1, f_2))$.

Proof. Let u be an edge in $E \setminus E(G')$. Let f_3 denote the map $C(f_1, f_2)$. Since G' is a spanning tree, by Lemma 110, the graph $(V, E(G') \cup \{u\})$ contains a cycle π which includes the edge u. Let π be the cycle of $(V, E(G') \cup \{u\})$ which includes the edge u. As G' is a MST of (G, f_1) and of (G, f_2), by Lemma 111, for any edge v in the cycle π, we have $f_1(v) \leq f_1(u)$ and $f_2(v) \leq f_2(u)$. Therefore, for any edge v in the cycle π, we have $min(f_1(v), f_2(v)) \leq min(f_1(u), f_2(u))$ and $max(f_1(v), f_2(v)) \leq max(f_1(u), f_2(u))$. Then, we should consider the three following cases:

1. If $min(f_1(v), f_2(v)) < min(f_1(u), f_2(u))$, then, by the hypothesis 4 on C, we have $C(f_1(v), f_2(v)) < C(f_1(u), f_2(u))$.

2. If $min(f_1(v), f_2(v)) = min(f_1(u), f_2(u))$ and $max(f_1(v), f_2(v)) < max(f_1(u), f_2(u))$, then, by the hypothesis 3 on C, we have $C(f_1(v), f_2(v)) \leq C(f_1(u), f_2(u))$.

3. If $min(f_1(v), f_2(v)) = min(f_1(u), f_2(u))$ and $max(f_1(v), f_2(v)) = max(f_1(u), f_2(u))$, then, by Lemma 114, we have $C(f_1(v), f_2(v)) = C(f_1(u), f_2(u))$.

Consequently, $C(f_1(v), f_2(v)) = f_3(v) \leq C(f_1(u), f_2(u)) = f_3(u)$. Hence, for any edge v in the cycle π, we have $f_3(v) \leq f_3(u)$. Thus, by Lemma 111, G' is a MST of (G, f_3). □

Lemma 116. *Let C be a positive function such that, for any a, b, c and d in $\{0, \dots, n-1\}$, we have:*

1. *$C(0, 0) = 0$; and*
2. *$C(a, b) = C(b, a)$; and*
3. *if $min(a, b) = min(c, d)$ and $max(a, b) < max(c, d)$ then $C(a, b) \leq C(c, d)$; and*
4. *if $min(a, b) < min(c, d)$ then $C(a, b) < C(c, d)$.*

Let f_1 and f_2 be the saliency maps of two hierarchies on V and let $\mathcal{B}$ be a binary partition hierarchy of (G, w) such that both f_1 and f_2 are one-side increasing for $\prec$. Then for any u in $E_\prec$, if u is not a watershed-cut edge for $\prec$, then $C(f_1, f_2)(u) = 0$.

Proof. Let u be an edge in $E_\prec$. If u is not an watershed-cut edge for $\prec$, then, by the second condition of Definition 18, we have $f_1(u) = 0$ and $f_2(u) = 0$. Therefore, $C(f_1, f_2)(u) = C(0, 0) = 0$. □

Lemma 117. *Let C be a positive function such that, for any a, b, c and d in $\{0, \dots, n-1\}$, we have:*

1. *$C(0, 0) = 0$; and*
2. *$C(a, b) = C(b, a)$; and*
3. *if $min(a, b) = min(c, d)$ and $max(a, b) < max(c, d)$ then $C(a, b) \leq C(c, d)$; and*
4. *if $min(a, b) < min(c, d)$ then $C(a, b) < C(c, d)$.*

Let f_1 and f_2 be the saliency maps of two hierarchies on V and let $\mathcal{B}$ be a binary partition hierarchy of (G, w) such that both f_1 and f_2 are one-side increasing for $\prec$. Let f_3 denote the map $C(f_1, f_2)$. Then, for any building edge u of $\mathcal{B}$, there exists a child R of R_u such that $f_3(u) \geq \vee\{f_3(v)$ such that R_v is included in $R\}$.

Proof. Since f_1 (resp. f_2) is one-side increasing for $\prec$, by the third condition of Definition 18, we have that, for any building edge u of $\mathcal{B}$, $f_1(u) \geq \vee\{f_1(v) \mid R_v \subseteq X\}$ (resp. $f_2(u) \geq \vee\{f_2(v) \mid R_v \subseteq X\}$) for a child X of R_u. We need to prove that, for any building edge u of $\mathcal{B}$, there is a child X of R_u such that $f_3(u) \geq \vee\{f_3(v) \mid R_v \subseteq X\}$. Let u be a building edge of $\mathcal{B}$ and let X and Y be the children of R_u. We should consider the following four cases:

1. If $f_1(u) \geq \vee\{f_1(v) \mid R_v \subseteq X\}$ and $f_2(u) \geq \vee\{f_2(v) \mid R_v \subseteq X\}$, then, for any building edge e such that $R_e \subseteq X$, we have $f_1(u) \geq f_1(e)$ and $f_2(u) \geq f_2(e)$. Let e be an edge edge such that $R_e \subseteq X$. Therefore, $min(f_1(e), f_2(e)) \leq min(f_1(u), f_2(u))$. If $min(f_1(e), f_2(e)) < min(f_1(u), f_2(u))$ then, by the hypothesis 4 on C, $C(f_1(e), f_2(e)) < C(f_1(u), f_2(u))$. Otherwise, we have $min(f_1(e), f_2(e)) = min(f_1(u), f_2(u))$. As $f_1(u) \geq f_1(v)$ and $f_2(u) \geq f_2(e)$, we have $\max(f_1(u), f_2(u)) \geq \max(f_1(e), f_2(e))$. If $\max(f_1(u), f_2(u)) = \max(f_1(e), f_2(e))$ then, by Lemma 114, $C(f_1(u), f_2(u)) = C(f_1(e), f_2(e))$. Otherwise, we have $\max(f_1(u), f_2(u)) > \max(f_1(e), f_2(e))$ and then by hypothesis 3 on C, we have $C(f_1(u), f_2(u)) \geq C(f_1(e), f_2(e))$. Thus in all cases we have $C(f_1(u), f_2(u)) \geq C(f_1(v), f_2(v))$, and by definition of f_3: $f_3(u) \geq f_3(e)$. Therefore, $f_3(u) \geq \vee\{f_3(v) \mid R_v \subseteq X\}$.

2. If $f_1(u) \geq \vee\{f_1(v) \mid R_v \subseteq X\}$ and $f_2(u) \geq \vee\{f_2(v) \mid R_v \subseteq Y\}$, then we have to consider two cases: (i) $f_1(u) \leq f_2(u)$ and (ii) $f_1(u) > f_2(u)$.

   (i) Assume that $f_1(u) \leq f_2(u)$. Then $min(f_1(u), f_2(u)) = f_1(u)$. Let v be an edge such that $R_v \subseteq X$. By our assumption, we have $f_1(u) \geq f_1(v)$. Indeed, since f is a one-side increasing map, we can say that either $f_1(u) = f_1(v) = 0$ or $f_1(u) > f_1(v)$ because only the watershed-cut edges for $\prec$ have non-zero and pairwise distinct weights. If $f_1(u) = f_1(v) = 0$, this implies that neither u nor v are watershed-cut edges for $\prec$ and therefore $f_2(u) = f_2(v) = 0$, which implies that $f_3(u) = 0 \geq f_3(v) = 0$. Otherwise, let us assume that $f_1(u) > f_1(v)$. In this case, and as $min(f_1(u), f_2(u)) = f_1(u)$, we have $min(f_1(u), f_2(u)) > f_1(v)$, and thus $min(f_1(u), f_2(u)) > min(f_1(v), f_2(v))$. Then by hypothesis 4 on C, we have $C(f_1(u), f_2(u)) > C(f_1(v), f_2(v)$ which is equivalent to $f_3(u) > f_3(v)$. Therefore, we have $f_3(u) \geq \vee\{f_3(v) \mid R_v \subseteq X\}$.

   (ii) If $f_1(u) > f_2(u)$ then we can apply the same reasoning as in the case where $f_1(u) \leq f_2(u)$.

3. $f_1(u) \geq \vee\{f_1(v) \mid R_v \subseteq Y\}$ and $f_2(u) \geq \vee\{f_2(v) \mid R_v \subseteq X\}$. This case is symmetric to 2.

4. $f_1(u) \geq \vee\{f_1(v) \mid R_v \subseteq Y\}$ and $f_2(u) \geq \vee\{f_2(v) \mid R_v \subseteq Y\}$. This case is symmetric to 1.

Thus, we can conclude that, for any building edge u of $\mathcal{B}$, there exists a child R of R_u such that $f_3(u) \geq \vee\{f_3(v)$ such that R_v is included in $R\}$.

□

of Property 47. By Lemma 69, we can affirm that $(V, E_\prec)$ is a MST of both $(G, \Phi(\mathcal{H}_1))$ and $(G, \Phi(\mathcal{H}_2))$. Therefore, by Lemma 115, $(V, E_\prec)$ is a MST of (G, f_3) as well, which proves that the first condition for $\mathcal{QFZ}(G, f_3)$ to be a flattened hierarchical watershed of (G, w) holds true. The second and third conditions are the result of Lemmas 116 and 117, respectively. Therefore, $\mathcal{QFZ}(G, f_3)$ is a flattened hierarchical watershed of (G, w).

□

8.4.3 Proof of Property 48

(Property 48). *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two hierarchical watersheds of (G, w). Let C be a combining function such that:*

$$C(x,y) = \begin{cases} 0 & if\ x{=}0\ and\ y{=}0 \\ \frac{x^m y^m}{x^m + y^m} & otherwise \end{cases} \tag{8.3}$$

for $m \geq n$. The combination of $\mathcal{H}_1$ and $\mathcal{H}_2$ with C is a flattened hierarchical watershed of (G, w).

Let C be the function:

$$C(x,y) = \begin{cases} 0 & if\ \text{x=0}\ and\ \text{y=0} \\ \frac{x^m y^m}{x^m + y^m} & \end{cases} \tag{8.4}$$

where m is equal or greater than the number of minima n of (G, w). We want to prove that, for any a, b, c and d in $\{0, \ldots, n-1\}$:

1. $C(0,0) = 0$; and
2. $C(a,b) = C(b,a)$; and
3. if $min(a,b) = min(c,d)$ and $max(a,b) < max(c,d)$ then $C(a,b) \leq C(c,d)$; and
4. if $min(a,b) < min(c,d)$ then $C(a,b) < C(c,d)$.

Since $m \geq n$, we can prove that those fours statements hold true for any a, b, c and d in $\{0, \dots, m-1\}$.

The proof of the first and second statements are trivial. In order to prove the third and fourth statements, we state Lemmas 118 and 119.

Lemma 118. *Let $C(x, y) = \frac{x^m y^m}{x^m + y^m}$ and let a, b and d be natural numbers such that $a \leq b$, $a \leq d$ and $b < d$. Then $C(a, b) \leq C(a, d)$.*

of Lemma 118. If $a = 0$, then $C(a, b) = 0$ which is less than or equal to $C(a, d) = 0$. Otherwise, let us assume that $a > 0$. We will prove that $C(a, b) \leq C(a, d)$ by proving that $C(a, d) - C(a, b)$ is positive.

$$C(a, d) - C(a, b) \tag{8.5}$$

$$= \frac{a^m d^m}{a^m + d^m} - \frac{a^m b^m}{a^m + b^m} \tag{8.6}$$

$$= \frac{a^{2m} d^m + a^m b^m d^m - a^{2m} b^m - a^m b^m d^m}{(a^m + d^m)(a^m + b^m)} \tag{8.7}$$

$$= \frac{a^{2m} d^m - a^{2m} b^m}{(a^m + d^m)(a^m + b^m)} \tag{8.8}$$

$$= \frac{a^{2m}(d^m - b^m)}{(a^m + d^m)(a^m + b^m)} \tag{8.9}$$

The denominator of the fraction (A.6) is clearly positive and, since $d > b$, we can say that $d^m - b^m$ is positive as well. Therefore, $C(a, b) - C(c, d)$ is positive and, consequently, $C(a, b) \leq C(c, d)$. □

Lemma 119. *Let $C(x, y) = \frac{x^m y^m}{x^m + y^m}$ and let a, b, c and d be natural numbers in $\{0, \dots, m-1\}$ such that $a \leq b$ and $c \leq d$. If $a < c$ then $C(a, b) < C(c, d)$.*

Proof. Let us define the function $f_a(y) = \frac{a^m y^m}{a^m + y^m}$ where y is a natural number. We will compute the limit of $f_a(y)$ for y tending to infinity in order to find the greatest value $C(a, y)$ for any y.

$$\lim_{y\to\infty}\frac{a^m y^m}{a^m+y^m} \tag{8.10}$$

$$=\lim_{y\to\infty}\frac{\frac{a^m y^m}{y^m}}{\frac{a^m+y^m}{y^m}} \tag{8.11}$$

$$=\lim_{y\to\infty}\frac{a^m}{\frac{a^m}{y^m}+\frac{y^m}{y^m}} \tag{8.12}$$

$$=\frac{a^m}{0+1} \tag{8.13}$$

$$=a^m \tag{8.14}$$

Let c be a value in $\{0,\ldots,m-1\}$ such that $a<c$. We will prove that $C(c,d)$ is greater than $\lim_{y\to\infty}\frac{a^m y^m}{a^m+y^m}=a^m$. Since $a<c$, we have $c\geq a+1$. If we prove that this lemma holds for the case where $c=a+1$, we can infer by recurrence that it holds for any c greater than a. Therefore, we can simply prove that $C(a+1,d)$ is greater than a^m for any $d>a$ (because $d\geq c$ by hypothesis). By Lemma 118, given any value d' such that $a+1\leq d'$, we have that $C(a+1,a+1)\leq C(a+1,d')$. Since $a<d$, the minimal value of d is $a+1$. Given that $d=a+1$, we have $C(a+1,d)=C(a+1,a+1)=\frac{(a+1)^{2m}}{2(a+1)^m}=\frac{(a+1)^m}{2}$. Then we only need to prove that $a^m<\frac{(a+1)^m}{2}$ or that $a^m-\frac{(a+1)^m}{2}<0$.

$$a^m-\frac{(a+1)^m}{2} \tag{8.15}$$

$$=a^m-\left(\frac{a+1}{\sqrt[m]{2}}\right)^m \tag{8.16}$$

$$=\left(a-\frac{a+1}{\sqrt[m]{2}}\right)\left(a^{m-1}+a^{m-2}\left(\frac{a+1}{\sqrt[m]{2}}\right)+\right.$$

$$\left.\cdots+a\left(\frac{a+1}{\sqrt[m]{2}}\right)^{m-2}+\left(\frac{a+1}{\sqrt[m]{2}}\right)^{m-1}\right) \tag{8.17}$$

The equation (A.15) is obtained by the factorization of the equation (A.14). The sign of equation (A.15) is determined by the first term $\left(a-\frac{a+1}{\sqrt[m]{2}}\right)$ because the other terms are positive since a and m are natural numbers. Thus, in order to prove that (A.13) is negative, we only need to show that $\left(a-\frac{a+1}{\sqrt[m]{2}}\right)<0$.

$$\left(a - \frac{a+1}{\sqrt[m]{2}}\right) < 0 \tag{8.18}$$

$$\sqrt[m]{2}a - a - 1 < 0 \tag{8.19}$$

$$a(\sqrt[m]{2} - 1) < 1 \tag{8.20}$$

$$a < \frac{1}{\sqrt[m]{2} - 1} \tag{8.21}$$

Hence, we need to demonstrate that $a < \frac{1}{\sqrt[m]{2}-1}$. Since a is in $\{0, \ldots, m-1\}$, we know that $a < m$ and we can simply prove that $m \leq \frac{1}{\sqrt[m]{2}-1}$ or that $\sqrt[m]{2}m - m \leq 1$ or $m - \sqrt[m]{2}m \geq -1$. Let us define the real function h as follows:

$$h(x) = x - \sqrt[x]{2}x \tag{8.22}$$

$$= x(1 - \sqrt[x]{2}) \tag{8.23}$$

$$= x(1 - e^{\frac{\ln 2}{x}}) \tag{8.24}$$

We will show that $h(x) \geq -1$ for any x in $[1, +\infty[$. If this holds true in the continuous case, we can infer that it also holds true in the discrete case. Given that $h(1) = -1$, we can prove that $h(x) \geq -1$ for any x in $[1, +\infty[$ by showing that $h(x)$ is increasing in the interval $[1, +\infty[$. To that end, we will verify that the derivative of $h(x)$ is positive for any x in $[1, +\infty[$.

$$h'(x) = 1 - e^{\frac{\ln 2}{x}} - x\left(e^{\frac{\ln 2}{x}}\left(-\frac{\ln 2}{x^2}\right)\right) \tag{8.25}$$

$$= 1 - e^{\frac{\ln 2}{x}} + e^{\frac{\ln 2}{x}}\frac{\ln 2}{x} \tag{8.26}$$

To verify that $h'(x)$ is positive in the interval $[1, +\infty[$, we compute the limite of $h'(x)$ when x goes to $+\infty$ and its derivative $h''(x)$ of $h'(x)$.

$$\lim_{x \to +\infty} h'(x) = 1 - e^{\frac{\ln 2}{+\infty}} + e^{\frac{\ln 2}{+\infty}} \frac{\ln 2}{+\infty} \tag{8.27}$$

$$= 1 - e^0 + e^0 \times 0 \tag{8.28}$$

$$= 0 \tag{8.29}$$

$$h''(x) = -\left(e^{\frac{\ln 2}{x}} \left(-\frac{\ln 2}{x^2}\right)\right) + \left(-\frac{\ln 2}{x^2}\right) e^{\frac{\ln 2}{x}} \frac{\ln 2}{x} + \left(-\frac{\ln 2}{x^2}\right) e^{\frac{\ln 2}{x}} \tag{8.30}$$

$$= -\frac{(\ln 2)^2}{x^3} e^{\frac{\ln 2}{x}} \tag{8.31}$$

Therefore, we can affirm that $h''(x)$ is negative for any x in $[1, +\infty[$, which implies that $h'(x)$ is decreasing in the interval $[1, +\infty[$. Since $h'(x)$ is decreasing and the limit of $h'(x)$ going to infinity is zero, we can say that $h'(x)$ is positive for any x in $[1, +\infty[$. In addition, as $h(1) = -1$, this implies that $h(x)$ is increasing in the interval $[1, +\infty[$. This implies that $h(x) \geq -1$ and, therefore, $m - \sqrt[m]{2}m \geq -1$. This completes the proof that $C(a, b) < C(c, d)$.

□

♠

The wide literature on graph theory invites numerous problems to be modeled in the framework of graphs. In particular, clustering and segmentation algorithms designed this framework can be applied to solve problems in various domains, including image processing, which is the main field of application investigated in this thesis. In this work, we focus on a semi-supervised segmentation tool widely studied in mathematical morphology and used in image analysis applications, namely the watershed transform. We explore the notion of a hierarchical watershed, which is a multiscale extension of the notion of watershed allowing to describe an image or, more generally, a dataset with partitions at several detail levels. The main contributions of this study are the following:

- *Recognition of hierarchical watersheds:* we propose a characterization of hierarchical watersheds which leads to an efficient algorithm to determine if a hierarchy is a hierarchical watershed of a given edge-weighted graph.
- *Watersheding operator:* we introduce the watersheding operator, which, given an edge-weighted graph, maps any hierarchy of partitions into a hierarchical watershed of this edge-weighted graph. We show that this operator is idempotent and its fixed points are the hierarchical watersheds. We also propose an efficient algorithm to compute the result of this operator.
- *Probability of hierarchical watersheds:* we propose and study a notion of probability of hierarchical watersheds, and we design an algorithm to compute the probability of a hierarchical watershed. Furthermore, we present algorithms to compute the hierarchical watersheds of maximal and minimal probabilities of a given weighted graph.
- *Combination of hierarchies:* we investigate a family of operators to combine hierarchies of partitions and study the properties of these operators when applied to hierarchical watersheds. In particular, we prove that, under certain conditions, the family of hierarchical watersheds is closed for the combination operator.
- *Evaluation of hierarchies:* we propose an evaluation framework of hierarchies, which is further used to assess hierarchical watersheds and combinations of hierarchies.

In conclusion, this thesis reviews existing and introduces new properties and algorithms related to hierarchical watersheds, showing the theoretical richness of this framework and providing insightful view for its applications in image analysis and computer vision and, more generally, for data processing and machine learning.