UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Edward Jorge Yuri Cayllahua Cahuina

Algorithms for hierarchical graph-based image segmentation

Belo Horizonte
2023

Edward Jorge Yuri Cayllahua Cahuina

Algorithms for hierarchical graph-based image segmentation

Final Version

Dissertation presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Advisor: Arnaldo DE ALBUQUERQUE ARAÚJO
Co-Advisor: Jean COUSTY, Yukiko KENMOCHI,
Guillermo CÁMARA CHÁVEZ

Belo Horizonte
2023

Cayllahua.Cahuina, Edward Jorge Yuri

C385a Algorithms for hierarchical graph-based image segmentation [recurso eletrônico] / Edward Jorge Yuri Cayllahua Cahuina. 2023.
1 recurso online (130 f. il, color.): pdf.

Orientador Arnaldo de Albuquerque Araújo.
Coorientador Guillermo Camara Chávez
Coorientador Jean Cousty,
Coorientador Yukiko Kenmochi,

Tese (Doutorado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.
Referências: f. 122-130.

1. Computação – Teses. 2. Segmentação de imagens - Teses.3. Análise hierárquica – Teses. 4. Algoritmo incremental – Teses. I. Araújo, Arnaldo de Albuquerque. II. Camara Chávez, Guillermo. III. Cousty, Jean. V. Kenmoch, Yukiko. VI. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação. VII. Título.

CDU 519.6*73(043)

Ficha catalográfica elaborada pela bibliotecária Belkiz Inez Rezende Costa
CRB 6/1510 - Universidade Federal de Minas Gerais - ICEx

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIENCIAS EXATAS
PROGRAMA DE POS-GRADUAÇAO EM CIENCIA DA COMPUTAÇAO

FOLHA DE APROVAÇÃO

Algorithms for hierarchical graph-based image segmentation

EDWARD JORGE YURI CAYLLAHUA CAHUINA

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores(a):

PROF. ARNALDO DE ALBUQUERQUE ARAÚJO - Orientador
Departamento de Ciência da Computação - UFMG

Digitally signed by GUILLERMO CAMARA CHAVEZ:22758677830
Date: 2023.05.17 11:08:04 -03'00'

PROF. GUILLERMO CÁMARA CHÁVEZ - Coorientador
Departamento de Computação - UFOP

Prof. Jean Cousty - Coorientador
ESIEE - Universitè Gustave Eiffel

Profa. Yukiko Kenmochi - Coorientadora
Greyc Laboratory - CNRS - University of Caen Normandy

PROFA. BEATRIZ Marcotegui
Centre de Morphologie Mathématique - MINES - ParisTech

PROF. SÉBASTIEN Lefèvre
IRISA - Université Bretagne Sud

PROF. RONALDO FUMIO HASHIMOTO
Departamento de Ciência da Computação - USP

PROF. WILLIAM ROBSON SCHWARTZ
Departamento de Ciência da Computação - UFMG

PROF. SILVIO JAMIL FERZOLI GUIMARÃES
Departamento de Ciencia da Computação - PUC - MG

Belo Horizonte, 9 de maio de 2023.

Agradezco a Dios por todas sus bendiciones, a mis padres Jorge y Benilda, así como a toda mi familia, quienes me brindan su amor, constante apoyo y compañía.

Acknowledgments

I am very grateful to all my professors who guided me on this journey. I want to express my profound appreciation to my PhD advisors Arnaldo Araújo, Guillermo Cámara, Jean Cousty, and Yukiko Kenmochi. I would like to thank Professor Guillermo Cámara for his guidance when I was a master's student and for his encouragement to pursue the academic path. Thank you, Professor Arnaldo Araújo, for accepting me as your doctoral student at UFMG and for all your assistance in making my doctoral studies at ESIEE Paris, Université Gustave Eiffel possible. Your continuous support, even during the most challenging times of my doctorate, has been invaluable. Special thanks to Jean Cousty and Yukiko Kenmochi. You have played an essential role in the success of my PhD. Your encouragement, assistance, and the academic rigor you upheld have all contributed to my professional excellence. Your unwavering personal support, even during the vulnerable times of our collaboration, made it possible for me to ultimately achieve the successful conclusion of my PhD.

I wish to extend my gratitude to all the members of the jury: Beatriz MARCOTEGUI, Sébastien LEFÈVRE, Ronaldo HASHIMOTO, William SCHWARTZ, Silvio GUIMARÃES. Thank you for agreeing to review my manuscript and for generously contributing your time and expertise during my PhD defense. I also wish to thank all my colleagues and friends who were part of my doctorate studies at UFMG and at ESIEE.

I would like to acknowledge the Brazilian funding agency CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) for their financial support. Additionally, I wish to express my acknowledgment to the Peruvian agency CONCYTEC-PROCIENCIA for the grant awarded and their financial support.

Finally, I want to express my deepest gratitude to my family for their unconditional love and support, I could not have come this far without you.

Resumo

A segmentação de imagens é um problema aberto na visão computacional que tem sido extensivamente investigado por muitos anos. A tarefa de segmentar uma imagem envolve dividir os pixels de uma imagem em diferentes partes, onde cada parte representa uma característica distinta na imagem. Existem duas abordagens gerais que foram desenvolvidas para a segmentação de imagens: segmentação de imagem plana e segmentação de imagem hierárquica. Em uma segmentação de imagem plana, o algoritmo visa capturar em uma única partição todos os elementos distinguíveis na imagem. Os objetos de uma imagem são compostos de vários detalhes; em uma abordagem de segmentação de imagem plana, tais detalhes e seus relacionamentos não são considerados. Isso significa que não há estrutura ou ideia de composição de detalhes para os objetos da imagem. Por outro lado, a segmentação hierárquica de imagens aborda a natureza multi-escalar de uma imagem e produz uma representação hierárquica da imagem.

A literatura sobre algoritmos de segmentação plana de imagens é muito ampla e atualmente possui fundamentos teóricos mais estabelecidos em comparação com a segmentação hierárquica de imagens. Do ponto de vista teórico e algorítmico, existem poucos métodos bem compreendidos na segmentação hierárquica; consequentemente, existe uma brecha entre os estudos teóricos bem estabelecidos feitos para segmentação plana em comparação com a segmentação hierárquica, tal brecha nos dá uma grande oportunidade para pesquisas sobre os aspectos teóricos e algorítmicos da segmentação hierárquica.

Nesta tese, estudamos as noções teóricas e algorítmicas para um algoritmo de segmentação hierárquica de imagens. Examinamos e analisamos um método de segmentação de imagens planas baseado em grafos bem estabelecido e o estendemos para uma abordagem de segmentação de imagens hierárquica. As principais contribuições desta tese são as seguintes:

- *Um método de segmentação hierárquico de imagens baseado em grafos:* propomos um método no qual, similarmente à extensão de watersheds de segmentação plana para segmentação hierárquica, um nível da hierarquia resultante corresponde a uma instância do problema de segmentação plana baseado em grafos planos.

- *Uma base teórica para hierarquizar um método de segmentação de imagens baseado em grafos:* elaboramos um formalismo preciso para estudar as propriedades formais para hierarquizar um método de segmentação baseado em grafos. Como resultado desta base teórica, somos capazes de propor algoritmos eficientes e exatos para produzir um método de segmentação hierárquica de imagens.

- *Uma série de estratégias para a identificação de hiperparâmetros do método:* estudamos a identificação de hiperparâmetros de nosso método baseado em grafos hierárquicos e propomos novas estratégias para configurar esses hiperparâmetros.
- *Uma extensão da segmentação baseada em grafos original:* propomos uma generalização sobre a noção de medida de relevância originalmente usada pelo método baseado em grafos, tal generalização nos permite produzir hierarquias que dão mais importância a regiões que contêm certas características .
- *A demonstração de bom desempenho em uma avaliação prática:* validamos e mostramos que todas as nossas propostas têm um bom desempenho em situações práticas, utilizando um framework de avaliação especificamente proposto para hierarquias.

Em conclusão, esta tese fornece uma nova base teórica e algorítmica sobre a hierarquização de uma segmentação de imagens baseada em grafos. Nossas contribuições levam a algoritmos eficientes e exatos para calcular um método de segmentação hierárquica de imagens baseado em grafos, que é prático para seu uso em análise de imagens e visão computacional.

Palavras-chave: Segmentação de imagem. Análise hierárquica. Quasi-flat zone. Algoritmo incremental.

Abstract

Image segmentation is an open problem in computer vision that has been extensively investigated for many years. The task of segmenting an image involves dividing the pixels of an image into different pieces, where each piece represents a distinguished characteristic in the image. There are two general approaches that have been developed for image segmentation: flat-image segmentation and hierarchical image segmentation. In flat-image segmentation, the algorithm aims to capture in a single partition all the distinguishable elements in the image. The objects in an image are composed of several details; in a flat-image segmentation approach, such details and their relationships are not considered. This means there is no structure or idea of the composition of details for the objects of the image. On the other hand, hierarchical image segmentation tackles the multi-scale nature of an image and produces a hierarchical image representation.

The literature about flat-image segmentation algorithms is ample and currently has a more established theoretical background than hierarchical image segmentation. From a theoretical and algorithmic perspective, there are few methods well understood in hierarchical segmentation; consequently, there is a gap between the well-established theoretical studies made for flat segmentation compared to hierarchical segmentation, such gap gives us a great opportunity for research on the theoretical and algorithmic aspects of hierarchical segmentation.

This thesis studies the theoretical and algorithmic notions for a hierarchical image segmentation algorithm. We examine and analyze a well-established graph-based flat image segmentation method and we extend it to a hierarchical image segmentation approach. The main contributions of this thesis are the following:

- *A hierarchical graph-based image segmentation method:* we propose a method that, similarly to the extension of watersheds from flat to hierarchical, one level of the resulting hierarchy corresponds to one instance of the flat graph-based segmentation problem.

- *A theoretical background for hierarchizing a graph-based image segmentation method:* we elaborate a precise formalism to study the formal properties to hierarchize a graph-based segmentation method. As a result of this theoretical background, we are able to propose efficient and exact algorithms to compute a hierarchical image segmentation method.

- *A series of strategies for the identification of key hyper-parameters of the method:* we study the identification of key hyper-parameters of our hierarchical graph-based method and propose new strategies to set up these hyper-parameters.
- *An extension of measures for the original graph-based segmentation:* we propose a generalization on the notion of relevance measure originally used by the graph-based method. Thanks to our generalization, we are able to include new relevance measures in the hierarchical graph-based image segmentation method. The new relevance measures produce hierarchies that give more importance to regions that contain certain characteristics.
- *The demonstration of good performance in a practical evaluation:* we validate and show that all of our proposals have a good performance in practical situations, using an evaluation framework specifically proposed for hierarchies.

In conclusion, this thesis gives new theoretical and algorithmic background on the hierarchization of graph-based image segmentation. Our contributions lead to efficient and exact algorithms to compute a hierarchical graph-based image segmentation method, which is practical for its usage in image analysis and computer vision.

Keywords: Image segmentation. Hierarchical analysis. Quasi-flat zone. Incremental algorithm.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Image segmentation is an open problem in computer vision that has been extensively investigated for many years. The task of segmenting an image involves dividing the pixels of an image into different pieces, where each piece represents a distinguished characteristic in the image [Gonzalez and Woods, 2006].

Generic image segmentation has been widely studied in computer vision and image processing communities. These communities recognize the importance of image segmentation in giving an effective representation for solving mid-level and high-level vision tasks. There are several vision tasks whose solutions are based or are directly benefited by a reliable image segmentation.

There are many important benefits that can be gained using image segmentation, among which we can mention the following:

- the extracted segments are meaningful units that carry informative features such as shapes, textures, etc.;
- the number of segments is often significantly lower than the number of pixels in the original image, resulting in a more compact representation of the image, which eventually leads to the benefit of speeding up other processing techniques;
- the superpixel representation obtained from an image segment usually provides better coherency and robustness with respect to using the raw pixels from the image.

In this thesis, we study image segmentation in the context of computing a partition of an image. In this context, the input image is partitioned into disjoint regions. The number of output regions can be arbitrary and the union of these regions forms the whole image. Thus, the goal of image segmentation is to create meaningful and coherent regions that can be useful to represent objects or regions of interest within an image.

Image segmentation provides an intermediate representation of the image that is useful to solve more complex visual problems. Indeed, image segmentation is not the end goal but an intermediate step in performing a more complex visual analysis of an image, and higher-level visual tasks such as object recognition, scene understanding, or image interpretation. In this sense, there have been important advances in performing such tasks

leveraging from advances in deep learning techniques. One can refer to Convolutional Neural Networks (CNNs) [Krizhevsky et al., 2012], Fully Convolutional Networks (FCNs) [Long et al., 2015], and U-Net [Ronneberger et al., 2015], which have gained immense popularity. However, these techniques do not directly compute a partition of an image, they are directly interested in performing visual tasks such as classification, object recognition, etc.. Thus, in this thesis, we do not perform an exhaustive study of this type of approach, and we make a more comprehensive study of image segmentation approaches that compute a partition of an image.

There are two general approaches that have been developed for image segmentation: flat-image segmentation and hierarchical image segmentation. In the next two sections, we discuss the benefits and shortcomings of each of these approaches.

1.1 Flat-image segmentation

1.1.1 Image segmentation

A flat-image segmentation algorithm produces a single partition of the image into disjoint sets. In this single partition, the algorithm tries to capture all the distinguishable elements of the image [Gonzalez and Woods, 2006]. As an example, Figure 1.1(b) shows the resulting flat-image segmentation of Figure 1.1(a). In this case, it can be observed that the algorithm generated one partition capturing the general form of the child. We must also notice in Figure 1.1(b), that the resulting partition does not consider some visual details, *e.g.*, the eyes of the child or the lips, etc. If one would like to obtain another partition that includes such elements, the segmentation algorithm would need to be modified or tuned in order to produce the desired final segmentation. Flat-image segmentation has been widely popular and is usually used for segmenting a specific object of interest in the image.

(a) (b)

Figure 1.1: (a) An image of a child, and (b) a typical result of flat-image segmentation.

1.1.2 Milestones in flat-image segmentation

The literature about flat-image segmentation algorithms is very ample; nonetheless, we can identify some important milestones in this approach. The first approach is based on thresholding an image, this is the most basic idea of segmenting an image using a threshold value. The Otsu algorithm [Otsu, 1979] is the most iconic of these algorithms, it aims to automatically select an optimal threshold value for separating objects of interest in an image from the background.

The watershed algorithm considers the input image as a topographical relief and the intensity values of a pixel is the elevation at such point. The algorithm identifies the catchment basins in this relief, the lines that separate different catchment basins are called watersheds and they are able to segment the image into different regions. The watershed algorithm was introduced by Digabel and Lantuéjoul [1978], Beucher and Lantuéjoul [1979] for image segmentation, later popularized by Vincent and Soille [1991], and is nowadays commonly considered in edge-weighted graphs as introduced in [Cousty et al., 2009].

The min cut algorithm [Boykov et al., 2001] considers the graph based representation of an image where each vertex is a pixel of an image, a set of terminal points represent the object of interest and the algorithm finds the cut of minimum weight for this set of terminal points, this leads to a segmentation of the foreground (terminal points) from the background of the image.

Another important graph based segmentation algorithm was proposed by Felzenszwalb and Huttenlocher [2004], they proposed a predicate to find the evidence of a boundary between regions. This is a graph-based method (GB), which is able to capture perceptually important groupings while also being very efficient.

Random walks [Grady, 2006] have also been proposed for image segmentation,

where the user defines a number of seeds that belong to the objects of interest. Then, the algorithm computes for all unseeded pixels the probability that a random walker starting at the pixel position reaches a seeded pixel. The final segmentation is obtained by selecting for each unseeded pixel the most probable seed destination.

Normal cuts were introduced by Jianbo Shi and Malik [2000], in this type of segmentation an image is treated as a graph partitioning problem. In order to segment the graph, they proposed a global criterion called normalized cut. This criterion aims to capture the dissimilarity between different regions in the graph as well as the similarity within the regions.

Shortest path is another approach for image segmentation in the framework of edge-weighted graphs. In this approach, the algorithm labels a pixel as foreground if there is a shorter path from such pixel to a foreground seed compared to any background seed. This approach was first popularized by Bai and Sapiro [2007], and variations based on this approach have also been proposed in [Criminisi et al., 2008, Falcao et al., 2004].

1.2 Hierarchical image segmentation

1.2.1 A hierarchical approach

In a flat-image segmentation approach, the algorithm aims to capture in a single partition all the distinguishable elements in the image. Nonetheless, the objects in an image are composed of several details; in a flat-image segmentation approach, such details and their relationships are not considered. In order to capture such details, the user would have to tune the algorithm in order to produce the desired segmentation. Figure 1.2 aims to illustrate such procedure, it shows a set of resulting segmentations capturing different levels of details from the face of a child, we can appreciate that from Figure 1.2(b) to Figure 1.2(f), many details from the face of the child are captured; such as the lips, the eyes, the hair, etc. In order to obtain them, one could perform several image segmentations using a classical flat-image segmentation algorithm.

However, flat-image segmentation does not compute the relation of these details with respect to the object they compose. This means that there is no structure or any idea of the composition of details for the objects of the image. Having such a notion of composition would allow the user to obtain image segmentations from different points of view of the scene. Indeed, this goes in accordance to the multi-scale nature of objects in

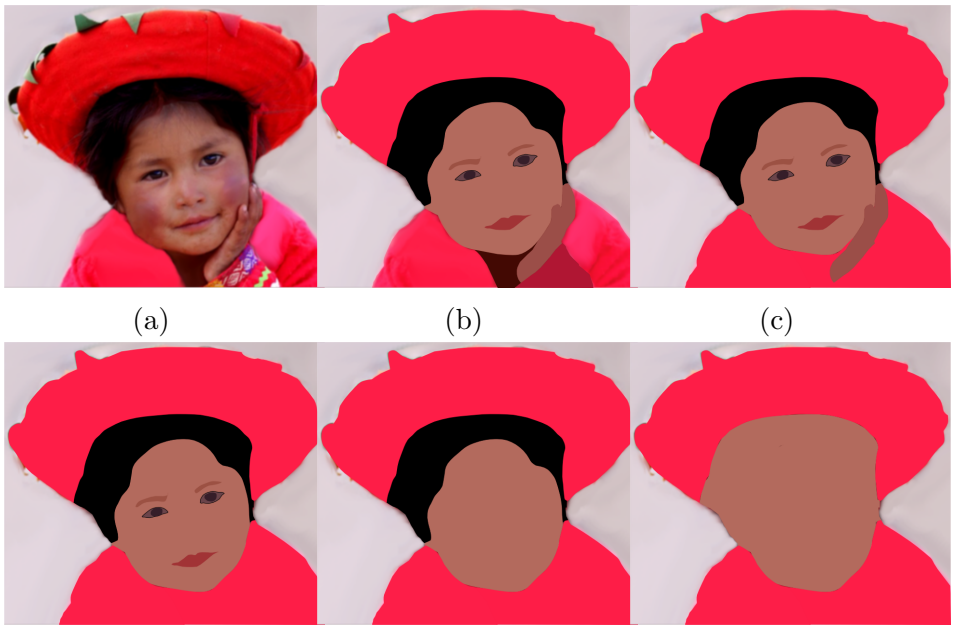


Figure 1.2: Figure 1.2(a) The image of a child from Figure 1.1(a), Figures 1.2(b)-(f) show a set of image segmentations capturing different levels of details for the child.

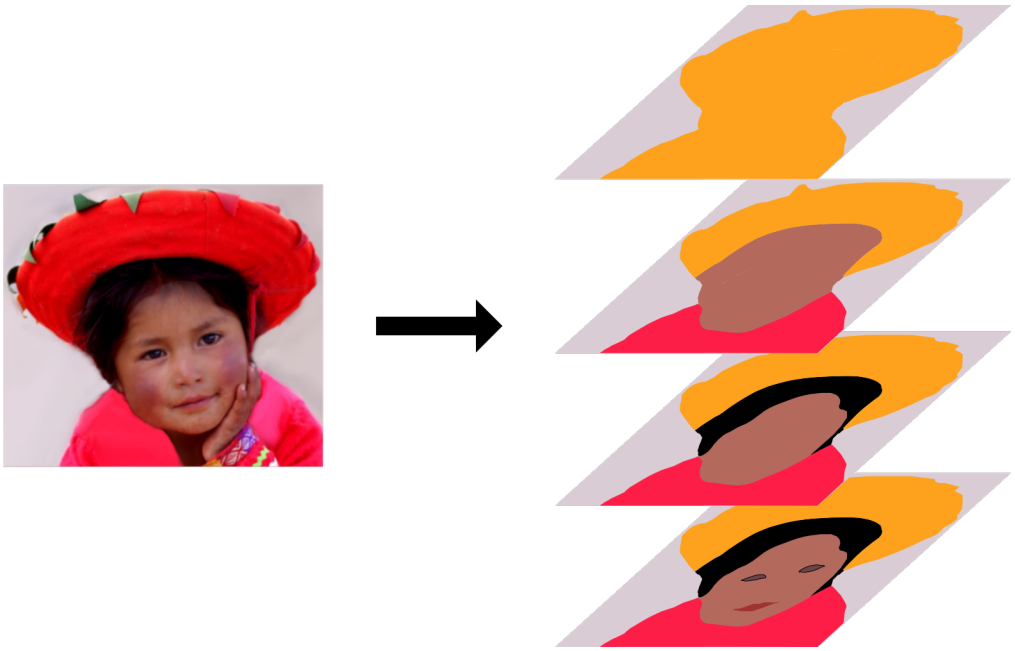


Figure 1.3: A hierarchy of image segmentations. An image can be decomposed into its objects and those objects into their sub-parts.

an image [Chen et al., 2016]. In a hierarchical image segmentation approach, we are able to capture in a single structure the various details of the objects in the image at multiple scales of interest. Thus, the image can be decomposed into its objects, and each object into their parts (see Figure 1.3).

In the hierarchical image segmentation approach, the user has the possibility to browse over different segmentations using only one structure. The user is also able to

decide which details are more relevant than others and get image segmentations according to their interest. This approach enables computer vision tasks to have a broader number of possibilities for analysis. Such remark has also been acknowledged by different researchers that consider image segmentation as an inherently multi-scale problem [Chen et al., 2016]. This multi-scale nature can be captured in a hierarchical image representation.

The theoretical foundations of hierarchical image representations were introduced by Serra [2006] and Ronse [2008]. Their work introduce us into the notion of segmentations and partitions and how the lattices of these partitions allow the construction of hierarchies. According to this notion, a hierarchy is an increasing sequence of partitions; this sequence starts from a partition of singletons and goes up to the greatest element of the lattice.

As a multi-scale structure, a hierarchy should preserve spatial consistency of all regions and capture the intra-connectivity of these regions. Formally, according to Guigues et al. [2006], a hierarchy should satisfy two important principles:

1. the *causality principle* which states that a contour present at an observation scale k_1 should also be present at any scale k_2, such that $k_2 < k_1$,
2. the *location principle* which states that contours should neither move nor deform from one observation scale to another.

A hierarchy should comply with these principles to represent a valid multi-scale structure from which we can perform multi-scale analysis.

In order to visualize a hierarchy, one could use a dendrogram, but this is not the only representation. In their work, Najman and Schmitt [1996] introduced the notion of saliency maps to visualize a hierarchy of watersheds. A saliency map can be visualized as an image, and as such, is a friendly representation for a user. Saliency maps were also studied by Cousty et al. [2018] along with minimum spanning trees as hierarchical representations. Based on this study, Cousty et al. [2018] proposed a framework that formalizes the links between hierarchies, saliency maps, and minimum spanning trees; making it possible to create hierarchical methods that can operate in either of the three representations.

1.2.2 Milestones in hierarchical segmentation

In this subsection, we present some of the most important milestones in hierarchical segmentation.

We can say that Quad-trees were among the first hierarchies of segmentations. They were introduced by Tanimoto and Pavlidis [1975] as a pyramid of segmentations,

going from fine to coarser segmentations. In this work, the hierarchy is computed from a predefined image division where the content of the image is not considered.

Trees of Shapes were introduced by Monasse and Guichard [2000], they proposed an algorithm to decompose an image into connected components at a corresponding level; then, these components are structured into an inclusion tree. This tree represents the geometrical inclusion of its components. Such a tree is a partial hierarchical representation of the image by making the leaves correspond to basic information (e.g., local extrema) and the nodes represent the detected shapes.

Hierarchical watersheds were introduced in [Beucher, 1994, Najman and Schmitt, 1996, Meyer, 1996, 2000] and are now considered in edge-weighted graphs [Cousty and Najman, 2011, Najman et al., 2013, Perret et al., 2018]. This hierarchy is constructed by considering the watershed segmentation of an image. This segmentation follows the notion of drops of water flowing on a topographic surface; this water flowing partitions the space along topological ridges. This notion can be used in images if we consider an image as a topological relief and we control the water flowing according to a certain attribute. Then, by ordering segmentations according to such attribute, a hierarchy can be constructed.

Salembier and Garrido [2000] introduced binary partition trees (BPT) for image segmentation, given an initial set of regions the algorithm determines the order at which neighboring regions should merge. The merging procedure relies on a similarity criterion and determines at which scale the merging process stops. The final BPT structure represents an image at various scales; it also encodes the relations between regions and their corresponding neighboring regions. Later, Randrianasoa et al. [2018] proposed a strategy to incorporate different merging criteria (metrics) thus generalizing the BPT construction framework.

Minimum spanning trees (MST) have also been applied to solve image analysis problems, where early use of MST in mathematical morphology was promoted by Meyer [1996]. In this context, an image is represented as a graph. It was later studied by Cousty et al. [2013] and found equivalence between the MST of this graph and the hierarchy of partitions induced by this graph. Thus, one could compute a hierarchy of partitions of a graph (image) based on its MST. There is also work on the collection of minimum spanning trees for an image, which forms a minimum spanning forest; Meyer [1994] shows the link between minimum spanning forests (MSF) and flooding from marker algorithms, which is a popular algorithm for watershed segmentation. This link was extended in the form of watershed cuts and the separation induced by minimum spanning forests relative to the minima, which has been proved by Cousty et al. [2009].

A very comprehensive survey on several hierarchical representations was presented by Bosilj et al. [2018]. There is also work on how to efficiently compute these hierarchies in [Havel et al., 2019].

Hierarchical segmentation has also benefited from deep learning techniques, Arbe-

laez et al. [2011] introduced a supervised gradient estimator called globalized probability of boundary (GpB), which performs better than previous gradients (*e.g.*, gradients based on Euclidean distance in color spaces). Along with this gradient, they also provided a method for transforming any contour detector into a hierarchical region tree, the representation is called Ultrametric Contour Map (UCM) and is a dual representation of a hierarchy of partitions. The use of learned gradients before the hierarchical construction usually leads to better segmentations results, this was also attested by Perret et al. [2018], where another supervised gradient estimator called structured edge detector (SED) [Dollár and Zitnick, 2015] was able to perform better than other non-supervised gradients.

There is also work on learning techniques in how the hierarchical structure is constructed, one of the pioneers is Pont-Tuset et al. [2017] with Multiscale Combinatorial Grouping (MCG). They proposed a method that builds a multiresolution image pyramid; then, they propose a grouping strategy to combine the detected multiscale regions. This strategy uses a classifier that combines the multiscale features obtained from the regions into a single probability of boundary estimation, which leads to the hierarchical representation of the image.

Another work using learning techniques was proposed by Maninis et al. [2018] who introduce Convolutional Oriented Boundaries (COB). They proposed an end-to-end architecture that incorporates a convolutional neural network (CNN) for contour detection at different levels of resolution of an input image; a UCM is computed for each level and, following the idea of MCG, they propose an algorithm that combines the multiscale regions into a single hierarchical boundary map. COB has been able to produce state-of-the-art performance in several computer vision applications. Also, following an end-to-end architecture, Chierchia and Perret [2019] introduced an optimization framework that uses a cost function based on a min-max operator which can be used by a gradient descent algorithm. This type of optimization allows the neural network architecture to learn a hierarchical segmentation of an image.

On the hierarchization of a flat image segmentation We mentioned the GB method as a flat image segmentation method. The GB method has gained widespread popularity due to its demonstrated effectiveness and efficiency, which have been thoroughly investigated and validated by numerous researchers. Among them, Guimarães et al. [2012] proposed a hierarchization of the GB method, thus introducing a hierarchical graph based image segmentation method (HGB). Guimarães et al. [2012] was focused in proposing a practical algorithm to hierarchize the GB method (and any method that works under the same machinery). However, the properties of the HGB method and their potential extensions have not been thoroughly studied. Indeed, a more profound study of the HGB method would enable us to understand and inherit the effectiveness and efficiency gained from the GB method accurately. Furthermore, from this understanding, we can explore new

ideas to propose a novel theoretical framework for the HGB method that could lead to different and improved segmentation results. This would also result in a hierarchical image segmentation method that is well-understood in terms of both accuracy and complexity, similar to its flat segmentation counterpart, the popular GB method.

1.3 Problem

To this moment, there is no single segmentation approach that can give a perfect segmentation result for whatever input image. This is the main reason why the image segmentation task is still an open problem in the computer vision community. Several approaches have been proposed in the literature to tackle image segmentation; Flat-image segmentation is still an ill-posed problem as it is unable to capture the multiscale nature of an image. Nonetheless, the extensive research in this approach has led to a more established theoretical background, where many properties and formal theorems are defined. The resulting algorithms are also well understood in both correctness and complexity. Hierarchical image segmentation aims to capture the multiscale nature of an image, thus covering a broader spectrum of applications where they perform better than the flat-image segmentation approach. Nonetheless, from a theoretical and algorithmic point of view, there are few methods well understood in hierarchical segmentation. Thus, there exists a gap between the good and more established theoretical studies made for flat segmentation compared to hierarchical segmentation.

In this thesis, we study the theoretical and algorithmic notions for a hierarchical image segmentation algorithm. The main objective of this thesis is to extend the well established GB method to a hierarchical image segmentation approach. In this sense, our hierarchical image segmentation method pursues the following objectives:

- objective 1: the proposal of a method where, similarly to the extension of watersheds from flat to hierarchical, one level of the resulting hierarchy should correspond to one instance of the flat GB segmentation problem;
- objective 2: the elaboration of a precise formalism to study properties to hierarchize the GB method;
- objective 3: the resulting theoretical background should lead to efficient exact algorithms;
- objective 4: the identification of key hyper-parameters of the method and new strategies to set up these hyper-parameters;

- objective 5: the demonstration of a good performance in a practical evaluation;
- objective 6: the generalization on the notion of relevance measure originally used by the graph-based method and new relevance measures for the hierarchical graph-based image segmentation.

Some of the reasons we chose the GB method is because the method is simple to describe, this would allow us for a better understanding of its properties and its extension to the hierarchical case. Also, the GB method relies on well-known and studied graph notions like MST, which are well-defined and are currently exactly solvable with efficient algorithms. Furthermore, the GB method shows to work very well in practical cases. It must also be pointed out that this thesis relies on a preliminary work started by our team; such study about hierarchizing the GB method was started in 2012 [Guimarães et al., 2012, Guimarães et al., 2017].

1.4 Main contributions

The main contributions from this thesis are the following:

- contribution 1: The study of a method that extends GB in the sense of objective 1, where each level of the resulting hierarchy is an approximation of an instance of the flat segmentation problem; from this study, we formulated a precise approximation based on an optimization formulation and an induction formula;
- contribution 2: A generalization of the baseline method, we achieved this by introducing a non-linear regularization of the original merging criterion over scale;
- contribution 3: An exact algorithm to produce the result of the method were given, proved to be correct, its complexity was analyzed as well as its execution time in practice;
- contribution 4: Three key hyper-parameters were identified and discussed
  - Dissimilarity measure between regions
  - Scale regularization strategies
  - Regional attributes
- contribution 5: An evaluation performed on Pascal-VOC [Everingham et al., 2010, 2012], MS-COCO [Lin et al., 2014] datasets, etc., based on the evaluation framework

of [Perret et al., 2018] showing that with the best combination of hyper-parameters that we have considered, we achieved a good performance and we showed practical results in our evaluations.

1.5 Outline of the thesis

The remainder of this thesis is organized as follows. Chapter 2 presents the basic notions to understand the graph-based framework for hierarchical segmentation. Chapter 3 presents a hierarchical graph-based method for image segmentation (HGB method), this chapter aims to help the reader understand the theoretical and computational challenges of hierarchical segmentation and the research opportunities that we have identified. Chapter 4 presents a series of efficient algorithms for the HGB method relying on a region dissimilarity measure. In this chapter, we propose a new theoretical background that precisely formalizes the properties of the HGB method and that allows us to present efficient algorithms to compute exactly the HGB method. Chapter 5 introduces several strategies for scale regularization. In this chapter, we present an incremental contribution where new techniques for the selection of other interesting observation scale values lead to better segmentation results. Chapter 6 introduces the exploration of new regional attributes that can be computed for each region, this aims to extend the original HGB method. From our exploration, regions with certain characteristics emerge and prevail in the final hierarchy. Chapter 7 presents the concluding remarks obtained in this thesis and also gives ideas about the future direction of our research.

1.6 Publications

During our research, we have published the following articles as contributions to the literature:

[1] Edward Jorge Yuri Cayllahua Cahuina, Jean Cousty, Yukiko Kenmochi, Arnaldo de Albuquerque Araújo, Guillermo Cámara-Chávez, Silvio Guimarães. *Hierarchical segmentation from a non-increasing edge observation attribute.* Pattern Recognition Letters - Special issue on Hierarchical Representations, DOI 10.1016/j.patrec.2019.12.014,

vol. 131, 2020, pp 105-112.

[2] Edward Jorge Yuri Cayllahua Cahuina, Jean Cousty, Yukiko Kenmochi, Arnaldo de Albuquerque Araújo, Guillermo Cámara-Chávez, Silvio Guimarães. *Efficient algorithms for hierarchical graph-based segmentation relying on the Felzenszwalb-Huttenlocher dissimilarity.* International Journal of Pattern Recognition and Artificial Intelligence, DOI 10.1142/S0218001419400081, vol. 33, no. 11, 2019, pp 1-27.

[3] Edward Jorge Yuri Cayllahua Cahuina, Jean Cousty, Yukiko Kenmochi, Arnaldo de Albuquerque Araújo, Guillermo Cámara-Chávez, Silvio Guimarães. *A study of observation scales based on Felzenswalb-Huttenlocher dissimilarity measure for hierarchical segmentation.* Proceedings of the 21st International Conference on Discrete Geometry for Computer Imagery, DGCI, Paris, France, Lecture Notes in Computer Science, Springer, DOI $10.1007/978-3-030-14085-414$, vol. 11414, 2019, pp 167-179.

[4] Edward Jorge Yuri Cayllahua Cahuina, Jean Cousty, Yukiko Kenmochi, Arnaldo de Albuquerque Araújo, Guillermo Cámara-Chávez. *Algorithms for hierarchical segmentation based on the Felzenszwalb-Huttenlocher dissimilarity.* Proceedings of the International Conference on Pattern Recognition and Artificial Intelligence, ICPRAI, Montreal, Canada, ISBN no. 1 895193 06 0, 2018, pp 108-113.

1.7 Resources

During our research we have implemented several algorithms, all the codes are available at:

https://cayllahe.github.io/hgbcode

Chapter 2

Basic notions on hierarchical segmentation

2.1 Introduction

In this chapter, we present some of the most important notions to understand hierarchical image segmentation. We present several definitions, notations, and symbols that were introduced and used in previous works and that are important to mention. All these notions will be used during the following chapters of this thesis and will help the reader understand the contributions of this thesis. First, Section 2.2 aims to explain the basic notions to understand hierarchical graph-based image segmentation. The concepts introduced in this section allow us to understand a graph-based representation of an image and hierarchical analysis. We first give notions on hierarchies; then, we give the notions on a graph and connected partitions. From these notions, we introduce hierarchical representations such as minimum spanning trees (and their partition hierarchy [Cousty et al., 2013]), quasi-flat zone hierarchies [Meyer and Maragos, 1999], and saliency maps [Najman and Schmitt, 1996]. We also introduce an important framework, proposed by Cousty et al. [2013, 2018], that presents the bijection between hierarchical representations; such as minimum spanning trees, quasi-flat zone hierarchies and saliency maps. This bijection allow us to use any of these representations in order to handle hierarchies.

Then, in Section 2.3, we review the theoretical background on how to deal with hierarchies; in aspects related to how to post-process a hierarchy to obtain assessable results. Later, in Section 2.4, we present some ideas on how to evaluate hierarchies and some of the existing frameworks available to measure the performance of hierarchies in the context of image segmentation. Finally, in Section 2.5, we review some of the most important ideas presented in this chapter.

2.2 Basic notions on hierarchical segmentation

2.2.1 Hierarchies of partitions

Image segmentation is the partition of an image into disjoint regions. In hierarchical analysis, a hierarchy is a set of partitions in which we can define a relation. A partition is associated to a certain scale in the hierarchy. Intuitively, a hierarchy captures the horizontal relation between regions that belong to the same scale; at the same time, they also determine a vertical inclusion relation between regions at different scales. We now formally define these notions.

Given a finite set V, a *partition* of V is a set $\mathbf{P}$ of nonempty disjoint subsets of V whose union is V. Any element of $\mathbf{P}$ is called a *region* of $\mathbf{P}$. This also means that, if x is an element of V, there is a unique region in $\mathbf{P}$ that contains x.

Given two partitions $\mathbf{P}$ and $\mathbf{P}'$ of V, $\mathbf{P}'$ is said to be a refinement of $\mathbf{P}$, denoted by $\mathbf{P}' \preceq \mathbf{P}$, if any region of $\mathbf{P}'$ is included in a region of $\mathbf{P}$. The notion of refinement is very important to understand the inclusion relation that there must exist between partitions in a hierarchy.

Formally, a hierarchy on V is a sequence $\mathcal{H} = (\mathbf{P}_0, \ldots, \mathbf{P}_\ell)$ of partitions of V, such that $\mathbf{P}_{i-1} \preceq \mathbf{P}_i$, for any $i \in \{1, \ldots, \ell\}$.

From this formal definition, we can understand that a hierarchy aims to capture the image structure in terms of region relations, and trying to represent the most meaningful segments of the image at a certain scale.

2.2.2 Graph representation of an image

Graphs have proved to be an effective framework for image processing and analysis. An image is made of pixels, each of which has associated an integer coordinate. This means that we can obtain its spatial information in the image; each pixel also has associated color information. An image can be represented as a graph by considering each pixel as a vertex and by establishing a relation between these vertices, *i.e.*, the edges of the graph. It is also possible to associate an edge to a value that represents a measure between two pixels (data points). This value is referred to as the weight of an edge and leads to edge-weighted graphs.

(a) Image of 3 by 3 pixels

(b) A vertex for each pixel

(c) 4–adjacency graph around the red vertex

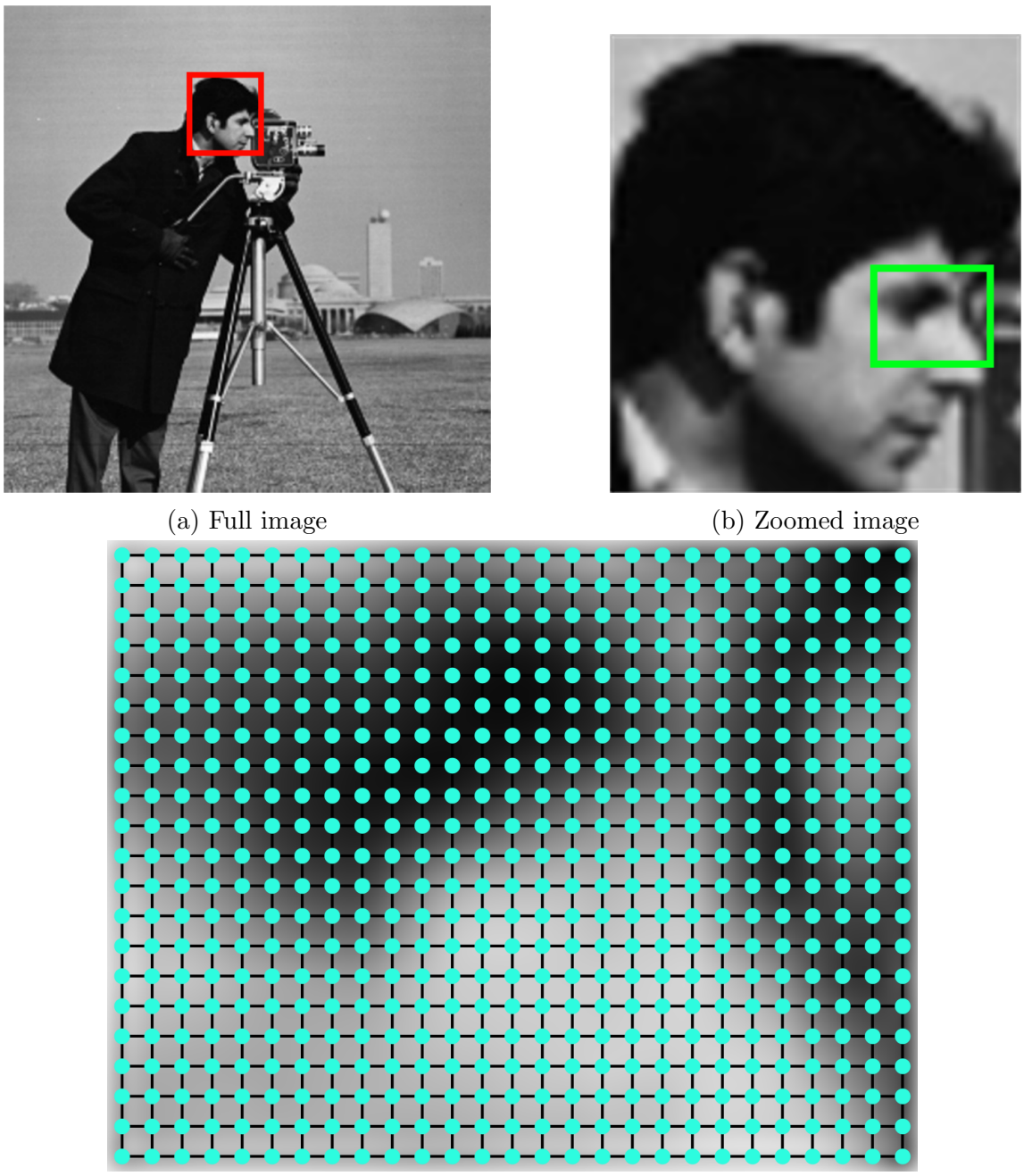
(d) 8–adjacency graph around the red vertex

Figure 2.1: Illustration of graph connectivity adjacency for an image.

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

In order to represent an image as a graph, researchers have used the pixel adjacency graph representation (see [Najman and Cousty, 2014] for an extended review on graph based image analysis in mathematical morphology). In this representation, the image is considered as a set of pixels represented by vertices and edges of the resulting graph linking neighboring pixels. Consequently, we need to define a pixel adjacency graph system to determine which edges have to be created. Such system determines the relation between neighboring pixels. In the case of a 2D image, we can consider a 4– or 8– adjacency system. Figure 2.1(a) shows a 4– adjacency and (b) shows an 8– adjacency. These figures illustrate how, using its corresponding integer coordinates, a pixel can be linked to the closest 4 or 8 pixels in its corresponding neighborhood.

Using these adjacency systems we can represent an image as a graph. For example, we can see in Figure 2.2 an image (a), a zoomed part from this image (b), and how each pixel of the image is considered as a vertex of a graph, whose edges are determined using the 4– adjacency relation (c).



(a) Full image (b) Zoomed image

(c) image graph representation, every pixel is an edge (interpolated)

Figure 2.2: Illustration of a graph representation for an image. Each pixel of the image is considered as a vertex of a graph, edges are determined using a 4– adjacency relation.

2.2.3 Gradient measures

In order to weigh the edges of a graph, techniques for image analysis have relied on a gradient measure. A simple way to compute a gradient measure is to use the colorimetric information from the two pixels sharing an edge. Every pixel has a value(s) associated with a particular color space; There are many color spaces, such as RGB, Lab, HLS, HSV, etc., each of which has its particular advantages and disadvantages [Gonzalez and Woods, 2006]. Given this colorimetric information between two pixels, one can compute the gradient measure as the Euclidean distance between the color spaces of the two pixels.

More advanced techniques have also been proposed to compute such gradients; for example, the structured edge detector (SED) proposed by Dollár and Zitnick [2015], which relies on a supervised gradient estimator. There is also a new breed of gradients based on deep learning techniques, like the one based on a U-shape architecture proposed by Liu et al. [2019].

The selection of a specific measure can have strong implications on the resulting hierarchy and thus final results of the segmentation [Perret et al., 2018].

2.2.4 Connected-component partition

Intuitively, a connected component represents a maximal set of vertices in which a path exists between any two vertices. For a formal definition we first introduce the definition of connected graph. Let x and y be two vertices of a graph G. A *path from x to y* in G is a sequence $(x_0, \ldots, x_m)$ of vertices of G such that $x_0 = x$, $x_m = y$ and $\{x_{i-1}, x_i\}$ is an edge of G for any i in $\{1, \ldots, m\}$. The graph G is *connected* if, for any vertices x and y of G, there exists a path from x to y. Let A be a subset of $V(G)$. The graph induced by A in G is the graph whose vertex set is A and whose edge set contains any edge of G made of two elements in A. If the graph induced by A is connected, then we say that A is *connected*. The subset A of $V(G)$ is a *connected component* of G if it is connected for G and maximal for this property, *i.e.*, for any subset Y of $V(G)$, if Y is a connected superset of X, then $Y = X$.

Having the notion of a connected component, we now introduce a connected-component partition. We denote by $\mathbf{C}(G)$ the set of all connected components of G. Note that $\mathbf{C}(G)$ is a partition of $V(G)$, which is called the *connected-component partition induced by* G. The set $[\mathbf{C}(G)]_x$ is the unique connected component of G that contains the element x.

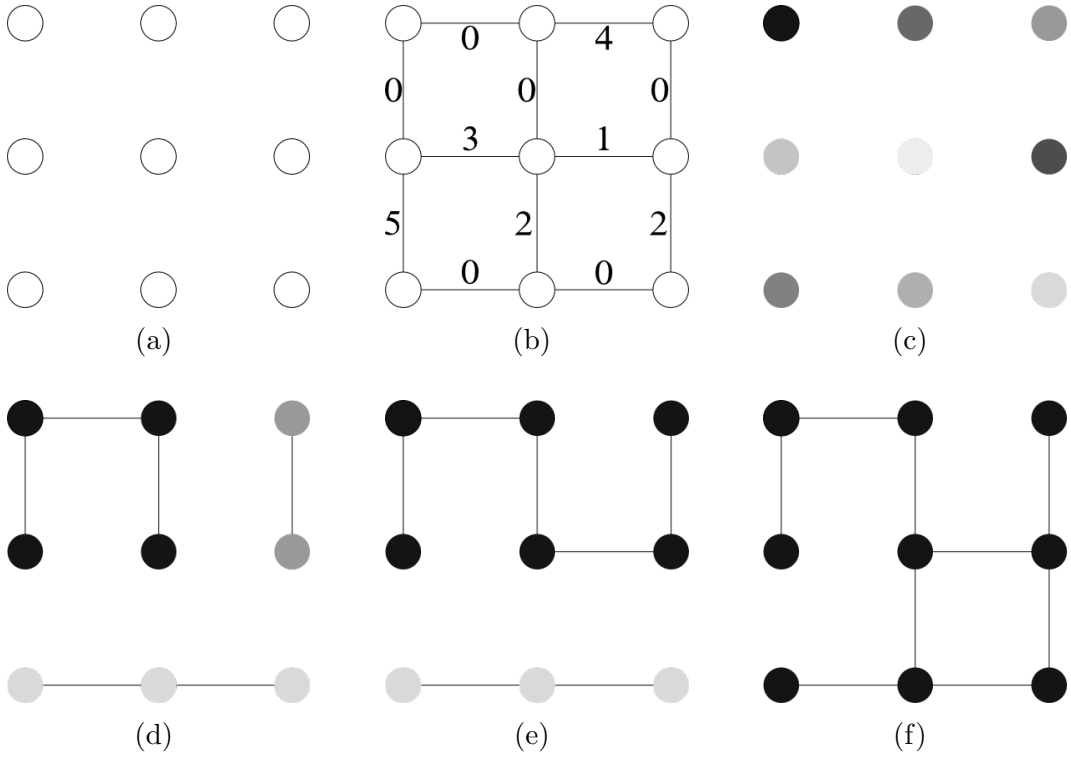


Figure 2.3: Illustration of connected-component partitions.

Figure 2.3 shows an illustration of connected-component partitions, where (b) shows a graph G, figures (c-f) show a connected-component partition for G, each colored region inside this partition is a connected component.

2.2.5 Quasi-flat zone hierarchies

The flat zone segmentation partitions the vertices of a non-negative edge-weighted graph.

The partition is obtained as the set of connected components of the graph whose vertices are those of the weighted graph and whose edges are those with weight below a given non-negative value. Quasi-flat zones are the basis for powerful hierarchical segmentation and filtering methods.

Given a graph $G = (V, E)$, let w be a map from E into the set $\mathbb{R}$ of real numbers. For any edge u of G, the value $w(u)$ is called the weight of u (for w), and the pair (G, w) is called an *edge-weighted graph*. We now explain how to make from an edge-weighted graph a series of connected-component partitions, which constitutes a hierarchy. Such hierarchy is called a quasi-flat zone hierarchy of (G, w) and the quasi-flat zone hierarchy transform is a bijection between the hierarchies and a subset of the edge-weighted graphs

(called the saliency maps).

Given an edge-weighted graph (G, w), let X be a subgraph of G and let λ be a value of $\mathbb{R}$. The *λ-level edge set* of X for w is defined by

$$w_\lambda(X) = \{u \in E(X) \mid w(u) < \lambda\}, \tag{2.1}$$

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

$$w^V_\lambda(X) = (V(X), w_\lambda(X)). \tag{2.2}$$

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

As we consider only finite graphs and hierarchies, the set of considered level values is reduced to a finite subset of $\mathbb{R}$ that is denoted by $\mathbb{E}$ in the remaining parts of this thesis. In order to browse the values of this set and to round real values to values of $\mathbb{E}$, we define, for any $\lambda \in \mathbb{R}$:

$$\begin{aligned} \mathrm{p}_{\mathbb{E}}(\lambda) &= \max\{\mu \in \mathbb{E} \cup \{-\infty\} \mid \mu < \lambda\}; \\ \mathrm{n}_{\mathbb{E}}(\lambda) &= \min\{\mu \in \mathbb{E} \cup \{\infty\} \mid \mu > \lambda\}; \text{ and} \\ \hat{\mathrm{n}}_{\mathbb{E}}(\lambda) &= \min\{\mu \in \mathbb{E} \cup \{\infty\} \mid \mu \geq \lambda\}. \end{aligned}$$

Let (G, w) be an edge-weighted graph and let X be a subgraph of G. The sequence, denoted by $\mathcal{QFZ}(X, w)$, of all λ-level partitions of X for w ordered by increasing value of λ, namely

$$\mathcal{QFZ}(X, w) = (\mathbf{C}(w^V_\lambda(X)) \mid \lambda \in \mathbb{E} \cup \{\infty\}), \tag{2.3}$$

is a hierarchy, called the *quasi-flat zone hierarchy of X for w*. Let $\mathcal{H}$ be the quasi-flat zone hierarchy of G for w. Given a vertex x of G and a value λ in $\mathbb{E}$, the region that contains x in the λ-level partition of the graph G is denoted by $\mathcal{H}^\lambda_x$.

Important notations and remarks. In the remaining parts of this thesis, the symbol G denotes a connected graph, the symbol w denotes a map from E into $\mathbb{R}$, and the symbol T denotes a minimum spanning tree of (G, w).

2.2.6 Minimum spanning tree (MST)

A minimum spanning tree poses the solution to a very popular problem in combinatorial optimization. In order to understand the problem, we first define the weight of a graph. Given an edge-weighted graph (G, w), the *weight of the graph G* is the sum of the weights of all edges u such that $u \in E(G)$. The problem states that given a connected

edge-weighted graph G, find a connected subgraph X such that the vertices of X are the same as G, namely $V(X) = V(G)$, and that the weight of X is minimal.

Minimum spanning trees are a popular technique used in image segmentation [Felzenszwalb and Huttenlocher, 2004, Wassenberg et al., 2009, Meyer, 1996]. They can successfully manage segmentations made up of several connected regions. Figure 2.4(a) shows an edge-weighted graph G, while (b) shows a spanning tree of graph G. Additionally, (c) shows a minimum spanning tree of G.

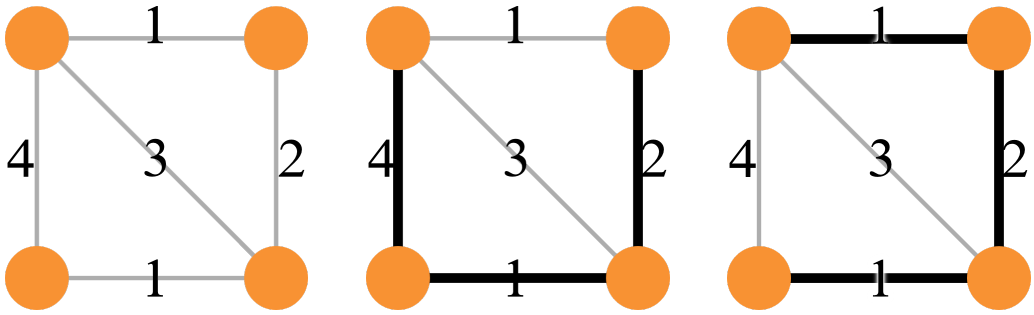


Figure 2.4: Illustration of a graph (a) and spanning tree for this graph (b), and minimum spanning tree for this graph (c).

2.2.7 Saliency maps

Saliency maps were first introduced to visualize hierarchies of watersheds in [Najman and Schmitt, 1996]. They are also known under the name of ultrametric contour maps [Arbelaez et al., 2011, Arbeláez and Cohen, 2006] and they are a very useful representation of hierarchical segmentations. They require an ultrametric distance measure to compute them, such as the one proposed by Meyer [2000]. In order to understand saliency maps; in this subsection, we introduce the notion of a graph cut (or cut). Intuitively, a cut in the hierarchy gives us the representation of the borders between regions in a hierarchy.

Let $\mathbf{P}$ be a partition of V, the *cut* of $\mathbf{P}$ (for G), denoted by $\phi_G(\mathbf{P})$, is the set of edges of G made of two vertices in different regions of $\mathbf{P}$:

$$\phi_G(\mathbf{P}) = \left\{ \{x, y\} \in E \mid [\mathbf{P}]_x \neq [\mathbf{P}]_y \right\}. \tag{2.4}$$

Let $\mathcal{H} = (\mathbf{P}_0, \ldots, \mathbf{P}_l)$ be a hierarchy on V. The map, denoted by $\Phi_G(\mathcal{H})$, is the saliency map of $\mathcal{H}$ from E to $\{0, \ldots, l\}$ such that the weight of any edge u for $\Phi_G(\mathcal{H})$ is the maximum value λ for which u belongs to the cut of $\mathbf{P}_\lambda$:

$$\Phi_G(\mathcal{H})(u) = \max \left\{ \lambda \in \{0, \ldots, l\} \mid u \in \phi_G(\mathbf{P}_\lambda) \right\}. \tag{2.5}$$

A naive algorithm to compute a saliency map was introduced by Cousty et al. [2018]. In order to compute the saliency map $s = \Phi_G(\mathcal{H})$ of a hierarchy $\mathcal{H} = (\mathbf{P}_0, \ldots, \mathbf{P}_l)$ induced by the graph $G = (V, E)$, we can follow two steps:

1. for every level λ of $\mathcal{H}$, compute the cut $\phi_G(\mathbf{P}_\lambda)$ of the partition $\mathbf{P}$ at level λ;
2. for every edge $u \in E$, set the value $\Phi_G(\mathcal{H})(u)$ to the maximum level λ such that u belongs to the cut $\phi_G(\mathbf{P}_\lambda)$.

For the first step, after computing the $\phi_G(\mathbf{P}_\lambda)$, we check if the edge u belongs to such cut $\phi_G(\mathbf{P}_\lambda)$. Then, in the second step, we can check for every edge u and every level λ if the edge u belongs to $\phi_G(\mathbf{P}_\lambda)$ and set the value $\Phi_G(\mathcal{H})(u)$ to the maximum level value such that this property holds true. Along with this algorithm, Cousty et al. [2018] also presented a more efficient algorithm that is based on computing the least common ancestor of two nodes in a tree representation of a hierarchy. Figure 2.5 shows an illustration of a saliency map of a hierarchy.

2.2.8 Bijection between hierarchies, quasi-flat zones and saliency maps

The work of Cousty et al. [2018] established the links of the maps that weight the edges of a graph $G = (V, E)$, the hierarchies on the vertex set of this graph $V(G)$ with the saliency maps on the edges of this graph $E(G)$, and the minimum spanning trees for the map that weight the edges of this graph.

Any edge-weighted graph induces a quasi-flat zone hierarchy and any hierarchy $\mathcal{H}$ can be represented by an edge-weighted graph whose quasi-flat zone hierarchy is precisely $\mathcal{H}$ (see [Cousty et al., 2018] for more details). This bijection indeed allows us to handle quasi-zone hierarchies through edge-weighted graphs. In other words, when an edge-weighted graph is created from an input image, the bijection signifies that this initial graph already possesses a quasi-flat zone hierarchy and that the procedure of hierarchical image segmentation can be interpreted as a modification of the initial hierarchical structure by changing edge weights on the graph.

Figure 2.6 shows the bijection that exists between quasi-flat zones hierarchies and saliency maps, and the links among edge-weighted graphs, minimum spanning trees, and quasi-flat zones hierarchies. Given an edge-weighted graph (G, w) and its corresponding minimum spanning tree (T, w), it has been shown by Cousty et al. [2018] that the quasi-flat zone hierarchy $\mathcal{QFZ}(T, w)$ of T for w is the same as the quasi-flat zone hierarchy $\mathcal{QFZ}(G, w)$ of G for w. One importance of such notion is that one can handle the quasi-flat zone hierarchy of G by its minimum spanning tree. Thus, in step of working on

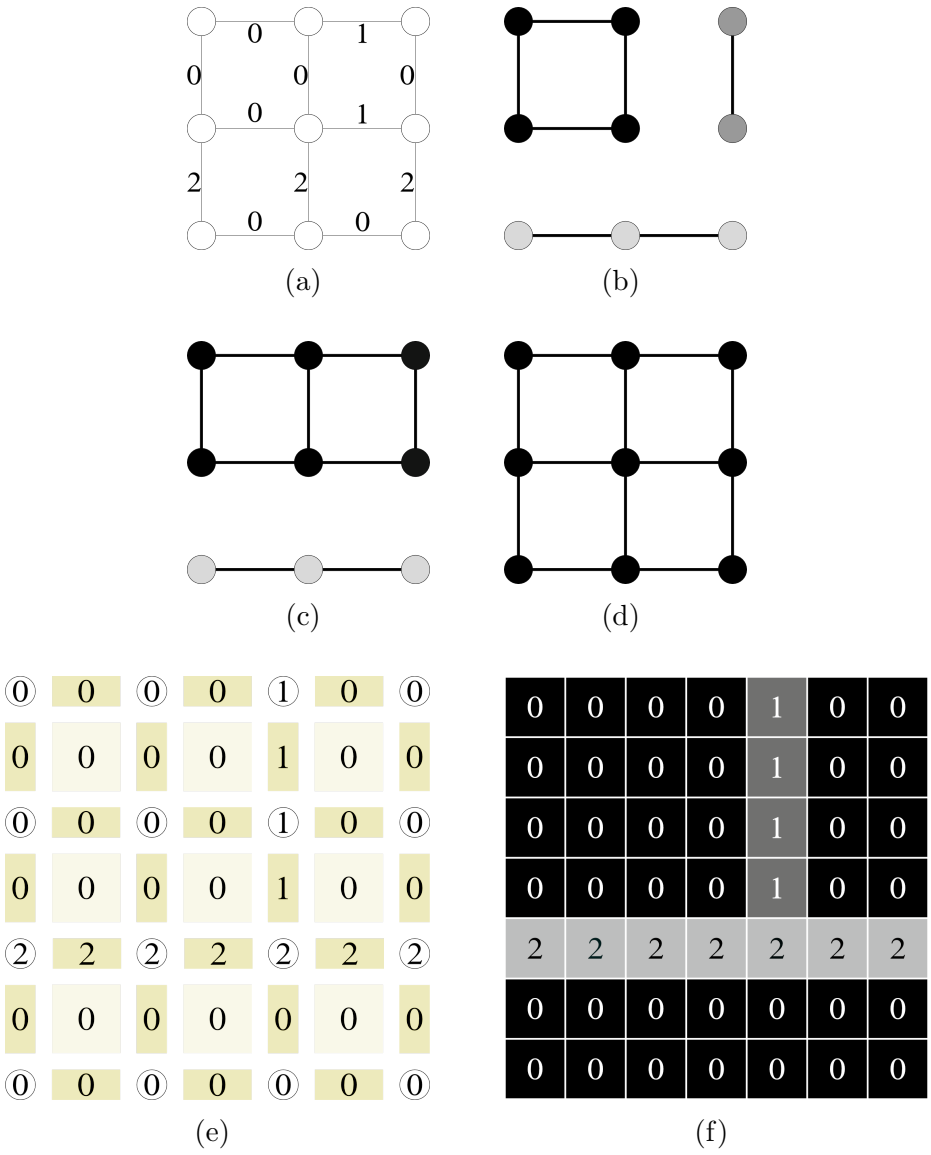


Figure 2.5: Illustration of a saliency map, (f) shows the map $s = \Phi_G(\mathcal{H})$ of the hierarchy $\mathcal{H}$ induced by the graph shown in (a), (b,c,d) show the 1-, 2-, and 3-level graphs of G for the computed map s, (e,f) shows a visualization of the saliency map when one considers the 4-adjacency graph. Figure extracted from Cousty et al. [2018].

the complete-graph-based representation of the hierarchy, we can be more efficient by working on its associated minimum spanning tree. Then, we can either go back to the quasi-flat-zones-hierarchy representation and use a saliency map to visualize the resulting hierarchy.

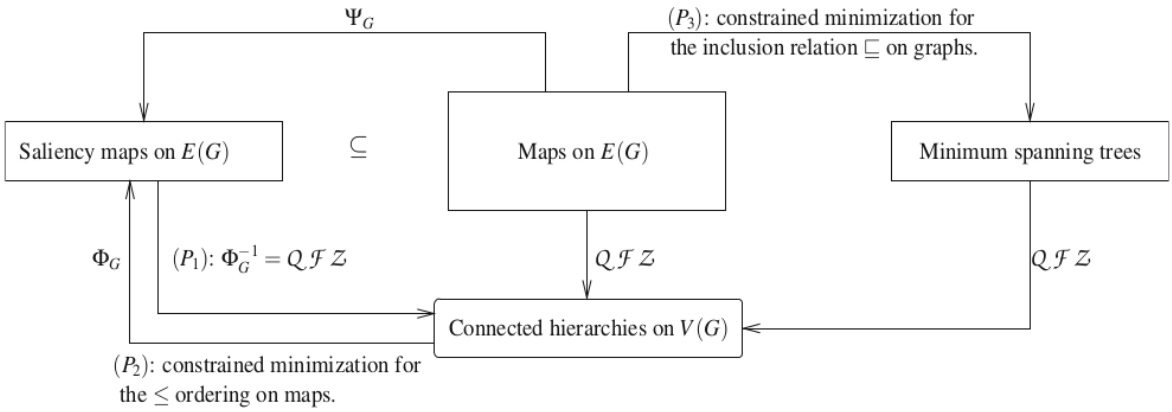


Figure 2.6: Links between the maps that weight the edges of a graph $G = (V, E)$, the hierarchies on the vertex set of this graph $V(G)$, the saliency maps on the edges of this graph $E(G)$, and the minimum spanning trees for the maps that weight the edges of this graph. Figure extracted from [Cousty et al., 2018].

2.3 Processing and transformation of hierarchies

The previous section presented several hierarchical representations, in the form of minimum spanning trees, quasi-flat zones, saliency maps and their equivalence. In this section, we illustrate the usefulness of these representations to the design of an efficient algorithm for removing insignificant regions in a hierarchy, as often required for post processing the result of hierarchical segmentation method [Perret et al., 2019]. The choice of which representation to use can be application-dependent [Salembier and Garrido, 2000, Bosilj et al., 2013], there exist many works on the transformation of hierarchies [Bosilj et al., 2013, Xu et al., 2017, Soille, 2008], that aim to process an initial hierarchy and transform it into a more suitable hierarchy for a particular problem. Equally important, is the simplification of a hierarchy, by removing non-relevant regions from it. it is always necessary that the most "interesting" and "useful" regions are present in the hierarchy; Therefore, a removal processing is necessary to detect and remove non-relevant regions from a hierarchy, this process is referred as hierarchy simplification.

The hierarchy simplification process is also an important step to obtain segmentation results that are practical and assessable in evaluation frameworks. In this process, it is important to ensure that the resulting transformation is a structure that preserves the hierarchical partition structure, *i.e.*, it is still a hierarchy. This type of processing is not unique to hierarchical analysis, Felzenszwalb and Huttenlocher [2004] also used a post-processing simplification step in their segmentation method to remove non-relevant regions; its hierarchical counterpart, namely the HBG method proposed by Guimarães et al. [2017], also used a similar simplification as a post-processing algorithm. This algo-

rithm is based on the framework of equivalence of hierarchies proposed by Cousty et al. [2018] and that we presented in Section 2.2.8. Guimarães et al. [2017] took an initial hierarchy and transformed it into its equivalent minimum spanning tree representation to remove regions whose size are smaller than a certain threshold. Perret et al. [2019] also used the same framework to propose an efficient algorithm to remove irrelevant regions from a hierarchy by exploiting the bijection between hierarchical representations. In this thesis, we also rely on the framework proposed by Cousty et al. [2018] and use the simplification algorithm proposed by Perret et al. [2019].

The main idea behind the algorithm proposed by Perret et al. [2019] is illustrated in Figure 2.7. The algorithm starts in an initial hierarchical representation $\mathcal{H}$; then, transforms $\mathcal{H}$ into its saliency map representation. In this domain, the algorithm is able to perform all the simplification processes in the weighted map w. After this processing, a new weighted map w' is obtained. Finally, the algorithm computes the quasi-flat zone hierarchy representation of w'. The result is the simplified hierarchy $\mathcal{H}'$. As follows, we will describe in more detail the simplification processing of this algorithm.

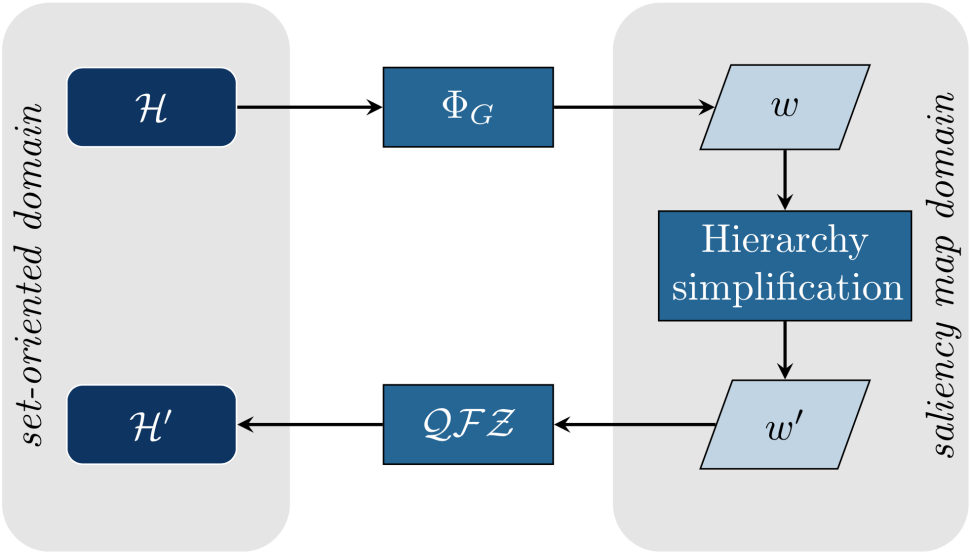


Figure 2.7: Simplification of a hierarchy by removing non-relevant regions from an initial hierarchy $\mathcal{H}$ and obtaining a resulting hierarchy $\mathcal{H}'$. Figure extracted from Perret et al. [2019].

Hierarchy simplification using an attribute criterion. In order to simplify a hierarchy $\mathcal{H}$ and remove non-relevant regions from $\mathcal{H}$, one of the key notions is to measure the importance of a region. Perret et al. [2019] proposed to use a regional attribute, such as the size of a region (called region area); other regional properties can also be used, for example, the contrast of a region. Once a regional attribute is defined, it is possible to compute the relevance of an individual region. It is also possible to set a threshold according to this measure and only keep regions whose level of relevance is greater than such threshold.

For a graph (G, w), Perret et al. [2019] proposed to use the minimal representation of its corresponding hierarchy H, which is a minimal weighted subgraph (T, w), where T is the minimum spanning tree of G. Given that we can handle hierarchies by their weight maps, the simplification algorithm aims to compute a weight map w' for the graph T. All regions in the associated hierarchy H' for (T, w') have a relevance greater than the defined threshold.

In addition, it can be said that any region in H' either existed previously in H (meaning that the region had a relevance greater than the threshold); or, it is a new region that was produced by merging a region whose relevance was lower than the threshold. Therefore, the merging procedure "removes" the non-relevant region by merging the region with another whose relevance is greater than the threshold. Then, for a hierarchy $\mathcal{H}$, a threshold value s and an attribute measure m, the algorithm follows the next steps to compute w':

1. Compute the minimum spanning tree T of G, w
2. for every edge $u = \{x, y\}$ in T
   a) get the largest region R_x in $\mathcal{H}$ that contains x but not y
   b) get the largest region R_y in $\mathcal{H}$ that contains y but not x
   c) if $m(R_x)$ and $m(R_y)$ are greater than s, then $w'(u) = w(u)$. Otherwise, $w'(u) = 0$

Steps 2(a) and 2(b) can be computationally expensive since it is necessary to browse the hierarchy (H) in each iteration. In order to efficiently perform Steps 2(a) and 2(b), Perret et al. [2019] proposed to use the binary partition tree by altitude ordering (BPTAO). For a graph $G = (V, E)$, a BPTAO represents a hierarchy of partitions of V, which can be efficiently obtained when we compute the minimum spanning tree T of G. Such efficient algorithms to obtain T and the BPTAO are presented by Najman et al. [2013].

After this computation, we obtain the weight map w'; from Figure 2.7 we can see that the next step is to compute the quasi-flat zone hierarchy from the graph (G, w') , which is the resulting simplified hierarchy $\mathcal{H}'$.

The simplified hierarchy $\mathcal{H}'$, is a processed hierarchy that is ready to be assessed by an evaluation framework. Each line in Figure 2.8 shows an input images, the computed Quasi-flat zone hierarchy for this image, its corresponding simplified Quasi-flat zone hierarchy; next, there is a watershed hierarchy and its corresponding simplified watershed hierarchy. We can observe that the resulting simplified hierarchies contain less regions than their input hierarchies and, more importantly, the most relevant regions are still present in the simplified hierarchies.

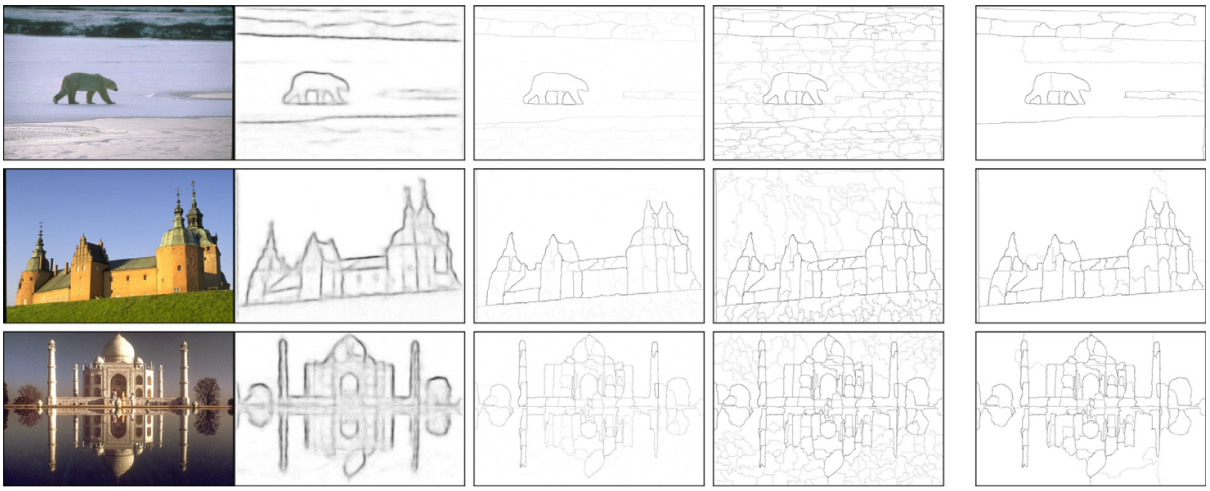


Figure 2.8: Simplification of hierarchies by removing non-relevant regions. A QFZ hierarchy and a Watershed hierarchy are simplified respectively and the corresponding saliency maps are shown for visual inspection. Figure extracted from Perret et al. [2019].

2.4 Evaluation of hierarchies

Assessing the quality of a hierarchy is not a simple task. In fact, there exist only a few evaluation frameworks in the literature [Randrianasoa et al., 2021]. One reason is that hierarchies are very complex structures from which we can extract several "optimal" cuts and obtain several segmentations. This leads to a combinatorial optimization problem where we need to get the most optimal cut to evaluate the hierarchy. Another difficulty is that most hierarchies are task-dependent; they are suited for a specific task where they can perform well but they may not perform as well in other tasks.

In order to evaluate hierarchies, one could perform a qualitative assessment by inspecting the saliency map of a hierarchy. In this visual inspection, one could check if the scale of a region is proportional to the brightness of the region contour. Although, this type of evaluation is not trivial and, as suggested by Perret et al. [2018], it can also be misleading. A quantitative assessment can be performed by evaluating the individual segmentations that can be obtained from the hierarchy. This is exploited by some evaluation frameworks, which take existing datasets for flat image segmentation which come with ground-truth segmentations; then, based on similarity measures, they assess if a segmentation can be obtained from the hierarchy that resembles the ground truth.

Two popular similarity measures between the two segmentations are the precision and recall scores. If the evaluation is performed along a sequence of segmentations, the precision-recall curve can be computed. This is the evaluation framework introduced by Arbelaez et al. [2011]. In order to evaluate hierarchies in the whole dataset, Arbelaez et al. [2011] also introduced the following two measures:

- OIS: refers to the optimal image scale, it computes an optimal horizontal cut in a hierarchy to get the best score;
- ODS: refers to the optimal data-set scale, it computes an optimal horizontal cut at the same level for all hierarchies in order to get the best score for the dataset.

One of the latest frameworks for the evaluation of hierarchies is proposed by Randrianasoa et al. [2021], which introduces a supervised framework to assess both the segmentation quality and the user's objective segmentation. Nonetheless, this framework is designed for Binary Partition Trees hierarchies. In order to assess hierarchies in the context of natural image segmentation, Perret et al. [2018] proposed a framework for performing quantitative assessments for hierarchical segmentation, particularly quasi-flat zones hierarchy and watershed hierarchies. Since the result of the HGB method is a quasi-flat zone hierarchy, this latter framework is the most appropriate for our work.

Perret et al. [2018] proposed new measures that can better evaluate a hierarchy in the context of natural images. The following new measures capture the potential gain of non-horizontal cuts compared to horizontal cuts:

- the Fragmentation–Optimal Cut score curve (FOC);
- the Fragmentation-Horizontal Cut score curve (FHC);
- the Object Detection Median (ODM).

The framework proposed by Perret et al. [2018] aims to assess the following two points:

- the quality of the "best" cuts or partitions in K regions appearing at the same scale (FHC measure) or at different scales (FOC measure) of the hierarchy. A cut is assessed with respect to human-provided ground truth and a segmentation quality measure (*e.g.*, Bidirectional Consistency Error-BCE). The measure is averaged for every K between 2 and twice the number of regions in the ground truth.
- the ease of finding a set of regions in the hierarchy corresponding to a semantic object when different levels of information is given on the position of the object: an object is considered easy to find when it can be well retrieved with few human-like-provided markers. Here, the retrieved object is a union of regions which, in general, does correspond to any cut of the hierarchy.

The assessment of the first aspect is summarized with a score called FOC+FHC, which sum up the performance of FHC and FOC. The evaluation of the second aspect is summarized with the ODM score. These scores will be useful for the assessments of our work.

2.5 Concluding remarks

This chapter introduced some of the most important notions to understand hierarchical image segmentation. We introduced the definitions, notations, and symbols that are used in this thesis. We also presented the theoretical background on how to treat hierarchies, in aspects related to how to post-process a hierarchy to obtain assessable results and how to measure their performance using an evaluation framework.

Chapter 3

A hierarchical graph-based segmentation method

3.1 Introduction

This chapter presents the hierarchical graph-based image segmentation method (HGB) proposed by Guimarães et al. [2017], which is based on the region-merging graph-based (GB) segmentation proposed by Felzenszwalb and Huttenlocher [2004]. The purpose of this chapter is to understand the properties of the HGB method and the machinery behind this framework. In this sense, in Section 3.2, we introduce the original GB method. We review some of the notions proposed by Felzenszwalb and Huttenlocher [2004] for their image segmentation method. We discuss the reasons why the GB method is a non hierarchical segmentation method. We also review one of the notions of the GB and HGB methods, which is related to a region dissimilarity measure, in Section 3.3. In practice, this measure allows us to determine if two regions of an image should be merged based on a region merging predicate.

Then, in Section 3.4, we review the notions of too-fine and too coarse partitions. Intuitively, any image segmentation method aims to partition the input image into meaningful regions. In this sense, if the segmentation method computes a too-fine partition, then, such partition creates an over-segmentation of the image. On the other hand, if the method computes a too-coarse partition, the resulting partition will not properly consider the details of the image, then an under-segmentation of the image is produced. Those notions for partitions were extended by Guimarães et al. [2017] to to not-too-fine/not-too-coarse hierarchies. The final part of Section 3.4 is dedicated to discuss how to compute such not-too-fine/not-too-coarse hierarchies.

In Section 3.5, we introduce the solution that Guimarães et al. [2017] proposed to compute a not-too-fine/not-too-coarse hierarchy and how it was incorporated into the HGB method. The main idea to answer this question is finding a maximal not-too-coarse hierarchy for the HGB method. Indeed, Guimarães et al. [2017] arguments that there is

no solution which is not-too-fine/not-too-coarse at the same time. The machinery behind this computation for finding a maximal not-too-coarse hierarchy is based on the notion of transformation of hierarchies proposed by Cousty et al. [2018].

All the basic notions presented in Sections 3.2, 3.4 and 3.5 allow us to present, in Section 3.6, the formal definition of the HGB method and its properties. We see that the core idea of the HGB method relies on transforming an initial hierarchy into a new hierarchy (transformation of hierarchies). In this transformation process, the structure of the initial hierarchy is reconstructed according to a region dissimilarity measure, so that a not-too-fine/not-too-coarse hierarchy is obtained. We discuss several points where we can elaborate a precise formalism of these properties. This study is important to achieve one of the objectives of this thesis, which is to propose a theoretical framework for the HGB method. Another importance of this chapter is to provide all the notions that help the reader understand the contributions made later in this thesis, which are theoretical and algorithmic (Chapters 4, 5 and 6). We finally discuss the most important ideas presented in this chapter in Section 3.8.

3.2 Graph-based image segmentation

Felzenszwalb and Huttenlocher [2004] introduced a graph-based segmentation method (GB) that compares two regions and evaluates whether there is boundary between the two of them. GB has gained a lot of popularity due to its simplicity and its good performance in practical scenarios. GB method uses a parameter allowing to control the scale of the resulting segmentation; larger values of the parameter is intended to lead to larger regions. However, as shown by Guimarães et al. [2017], GB is a non-hierarchical segmentation method. This means that the series of segmentations obtained when considering all possible values of the scale controlling parameter does not form a hierarchy, in the sense presented in Chapter 2 (see below). In the remaining of this section, we review in details the non-hierarchical behavior of GB method while, in the following sections of the chapter, we present the variation of GB method, called HGB, which was introduced in [Guimarães et al., 2017], and that is indeed a hierarchical segmentation method in the sense of Chapter 2.

While in Chapter 2, we introduced the notion of a hierarchy, here we introduce the notion of a level in a hierarchy as an "observation scale". In a hierarchy, two important principles should be satisfied: causality and location. A hierarchy satisfying the causality principle has the property that, the contour which exists at a level k_1 should also exist at any lower level k_2 in the hierarchy. In order to satisfy the location principle, the contours

in the hierarchy should be stable. This refers to the condition that a contour should not be deformed or displaced from one level to another. The level in the hierarchy can also be referred to as an "observation scale", and it relates to the level of simplification of the image at that level in the hierarchy.

As we saw in Chapter 2, hierarchies can be handled using many representations, from their minimum spanning trees, saliency maps, Quasi-flat zones, etc. [Cousty et al., 2018]. There are image segmentation algorithms whose analysis and results are based on minimum spanning tree, such as the ones from Felzenszwalb and Huttenlocher [2004], Nock and Nielsen [2004]. Nonetheless, the results from these algorithms do not produce a hierarchy. Furthermore, users are required to set and tune the parameter values if necessary. More in particular, the algorithm from Felzenszwalb and Huttenlocher [2004] used the parameter k that controls the level of simplification the user wants to achieve in the final segmentation. At a certain level of simplification k, Felzenszwalb and Huttenlocher [2004] used a region merging predicate to merge regions and produce a segmentation. We could think that by tuning this parameter value k as an "observation scale" we could get a hierarchy. however, Guimarães et al. [2017] noted the following violations:

1. As the parameter k increases, it can happen that the number of regions in the segmentation also increased. Such situation should not be possible if k was an "observation scale". Figure 3.1(b-d) shows the segmentation result from the algorithm proposed by Felzenszwalb and Huttenlocher [2004]. We are able to see that when k is set to 10000 the contours are different to the existing contours when k is set to 9000; thus, violating the causality principle.

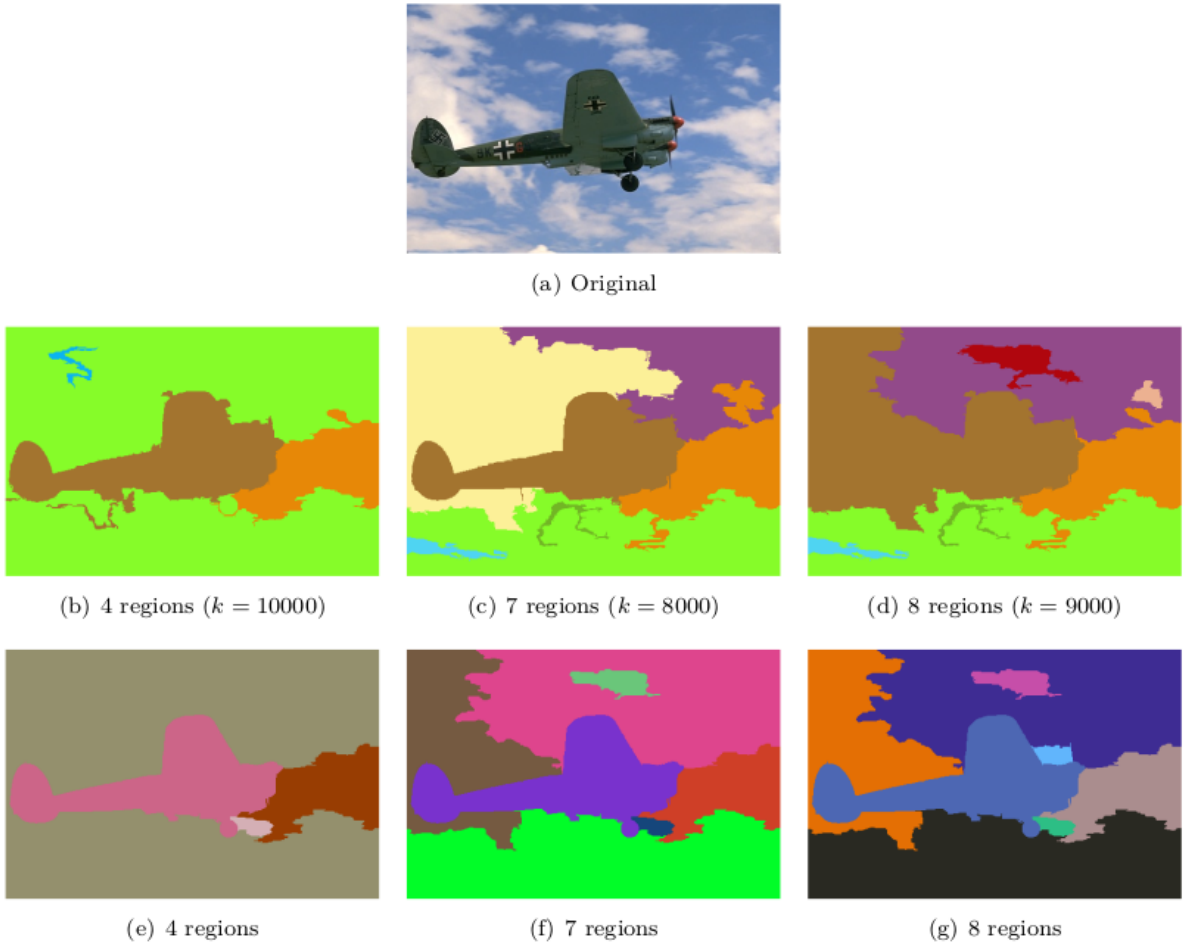
2. Contours are not stable from one scale to the other. See Figure 3.1(c) compared to Figure 3.1(d), the contours of the plane did not hold and then on 3.1(b) some of the contours present in 3.1(c) reappear. Thus violating the location principle.

3.3 Region dissimilarity measure for GB and HGB method

In the original GB method proposed by Felzenszwalb and Huttenlocher [2004], two adjacent regions of an image are merged based on the following region merging predicate.

Let R_1 and R_2 be two adjacent regions, the dissimilarity measure compares the so-called inter-component and within-component differences [Felzenszwalb and Huttenlocher, 2004]. The *inter-component difference* between R_1 and R_2 is defined by

$$\Delta_{inter}(R_1, R_2) = \min\{w(\{x, y\}) \,|x \in R_1, y \in R_2, \{x, y\} \in E(T)\},$$



(a) Original

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

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

Figure 3.1: Illustration of a non-hierarchical image segmentation method. Figures (b), (c), and (d) show that the number of regions is non-monotonic as scales increase in the method proposed by Felzenszwalb and Huttenlocher [2004]. In contrast, Figures (e), (f), and (g) show the monotonic behavior of the HGB method. (Image extracted from [Guimarães et al., 2017]).

while the *within-component difference* of a region R is defined by

$$\Delta_{intra}(R) = \max\{w(\{x, y\}) \,|x, y \in R, \{x, y\} \in E(T)\}.$$

It leads to the *observation scale of* R_1 *relative to* R_2, defined by

$$S_{R_2}(R_1) = (\Delta_{inter}(R_1, R_2) - \Delta_{intra}(R_1))\,|R_1|, \tag{3.1}$$

where $|R_1|$ is the cardinality of R_1. Then, a symmetric metric between R_1 and R_2, called the *observation scale dissimilarity between* R_1 *and* R_2, is defined by

$$D(R_1, R_2) = \max\{S_{R_2}(R_1), S_{R_1}(R_2)\}. \tag{3.2}$$

This dissimilarity is used to determine if two regions should be merged or not at a certain observation scale. In fact, the computation of this dissimilarity can be modified in order to capture other types of image features. This will be further studied in our thesis and we propose a generalization of such dissimilarity measure in Chapter 6.

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

An image segmentation method aims to partition it into meaningful regions; if the final result is a too-fine partition, such partition does not contain meaningful information and neither does a too-coarse partition. Ideally, we are interested in partitions that are neither too fine nor too coarse, Felzenszwalb and Huttenlocher [2004] aimed at producing such not-too-fine/not-too-coarse partition based on the merging predicate criterion that we introduced in the previous section. This section presents how Guimarães et al. [2017] handled this criterion in the HGB method and used it to produce hierarchies that are either not-too-fine nor not-too-coarse. We present the notion of too-fine/too-coarse partitions and then show how to extend this notion to hierarchies and therefore construct an optimal not-too-fine/not-too-coarse hierarchy.

3.4.1 Too-fine/too-coarse partitions

Let us consider a graph $G = (V, E)$, in which there is a partition P of V and a region pair $A, B \in P$. We say that the region pair A, B are *adjacent* if there exists an edge $u = \{x, y\} \in E$ such that $x \in A$ and $y \in B$. Let us also consider a dissimilarity measure D defined for adjacent regions and an observation scale λ. Then, the following definitions about the partition P are considered:

- P is a *too-fine partition* at level λ if there exists an adjacent region pair $A, B \in P$ such that $D(A, B) \leq \lambda$. This notion implicates that there exists an adjacent region pair A, B that, according to the dissimilarity measure D and that $D(A, B) \leq \lambda$, the two regions should have been merged and not separated. If P is a too-fine partition, then we can deduce that we have an over-segmentation.

- P is a *too-coarse partition* at level λ if there exists a proper refinement of P that is not too fine at level λ. This notion implicates that P is a too-coarse partition at level λ, if there exists a partition P' obtained from splitting a region of P and P' is not a too-fine partition; then, we can deduce that we have an under-segmentation.

Originally, Felzenszwalb and Huttenlocher [2004] aimed to compute a partition that is neither too-fine nor too-coarse for a given parameter k. Then, in order to propose a framework to hierarchize this type of algorithm, Guimarães et al. [2017] extended this notion to hierarchies.

3.4.2 Too-fine/too-coarse hierarchies

Guimarães et al. [2017] developed the notion of too-fine/too-coarse partitions to too-fine/too-coarse hierarchies. As Felzenszwalb and Huttenlocher [2004] were interested in partitions that were neither too-fine nor too-coarse, Guimarães et al. [2017] aimed at finding a hierarchy that is neither too-fine nor too-coarse. In this sense, let us consider a complete hierarchy $H = (P_\lambda \mid \lambda \in \mathbb{E})$ that is connected for G. We say that a *hierarchy* $\mathcal{H}$ *is not too-fine* if, for any $\lambda \in \mathbb{E}$, the partition P_λ is not too-fine at level λ. Correspondingly, a *hierarchy* $\mathcal{H}$ *is not too-coarse* if, for any $\lambda \in \mathbb{E}$, the partition P_λ is not too-coarse at level λ.

Ideally, we are only interested in producing hierarchies which are neither too-fine nor too-coarse. Nonetheless, generating such hierarchies is not a straightforward task. For example, as noted by Guimarães et al. [2017], the partitions produced by using the dissimilarity measure of Felzenszwalb and Huttenlocher [2004] do not lead to such neither too-fine nor too-coarse hierarchy. Indeed, as we can see from Figure 3.1, tuning the parameter k as an observation scale can lead to either a too-fine or too-coarse partition. Thus, Guimarães et al. [2017] focused on hierarchies that are not too-coarse.

Guimarães et al. [2017] noted that not-too-coarse hierarchies always exist. For example, let us consider a hierarchy $H = (P_\lambda \mid \lambda \in \mathbb{E})$ where every partition P_λ is made of singleton regions; we can say that such hierarchy is not too-coarse since there does not exist a proper refinement of a partition P_λ, whatever λ is considered. Furthermore, using any type of dissimilarity measure we can always aim to get not too-coarse hierarchies. In general, there can exist many not too-coarse hierarchies for a graph and we have to produce one based on a certain criterion. This is a very complex task, and in this thesis, we are not considering the challenge of producing not too-fine hierarchies.

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

In order to find a not-too-coarse/not-too-fine hierarchy, Guimarães et al. [2017] proposed to search for such hierarchy as an optimization problem. To this end, the first step was to propose a lattice structure for hierarchies in which to perform this optimization. The hierarchies used in this thesis are part of the techniques proposed in mathematical morphology for hierarchical analysis. The algebraic base for mathematical morphology is the lattice structure on which Guimarães et al. [2017] perform the optimization. A lattice is a partially ordered set such that for any group of elements, we can always find a

least upper bound and a greatest lower bound called supremum and infimum, respectively [Serra, 2006]. Consequently, Guimarães et al. [2017] defined the notions of infimum and supremum for hierarchies; this was also done by extending the notions for partitions.

Let us consider two partitions P and P', if P is a refinement of the partition P', then we say that P *is smaller* than P'. Correspondingly, we can say that P' *is larger* than P. Intuitively, if P' *is larger* than P, then P' is coarser than P. Then, we can define the lattice structure made of all partitions of V and the order relation "is larger than". From this lattice, the *infimum* of two partitions is the largest partition which is smaller than the two original partitions. And, the *supremum* of two partitions is the smallest partition which is larger than the two original partitions [Ronse, 2014, Serra, 2006].

The above notions allow us to define the "is larger than" in hierarchies. A hierarchy $\mathcal{H}$ *is larger than* another hierarchy $\mathcal{H}'$ if, at every level λ, the partition $\mathcal{H}_\lambda$ is larger than the partition $\mathcal{H}'_\lambda$. Consequently, the *infimum* of two hierarchies relates to the infimum of the partitions of two hierarchies, at every level. Correspondingly, for the *supremum* of two hierarchies, at every level. Having defined a lattice structure for hierarchies, it is then possible to propose an optimization in this space. Guimarães et al. [2017] defined it as follows: let $\mathbb{H}$ be a set of hierarchies, we say that an element $\mathcal{H}$ of $\mathbb{H}$ is minimal in $\mathbb{H}$, whenever, for any hierarchy $\mathcal{G}$ of $\mathbb{H}$ such that $\mathcal{G}$ is smaller than or equal to $\mathcal{H}$, we have $\mathcal{H} = \mathcal{G}$. If $\mathcal{H}$ is minimal in $\mathbb{H}$, we also say that $\mathcal{H}$ is a smallest hierarchy among the hierarchies of $\mathbb{H}$. Correspondingly, we say that an element $\mathcal{H}$ of $\mathbb{H}$ is maximal in $\mathbb{H}$, whenever, for any hierarchy $\mathcal{G}$ of $\mathbb{H}$ such that $\mathcal{G}$ is larger than $\mathcal{H}$, we have $\mathcal{H} = \mathcal{G}$. If $\mathcal{H}$ is maximal in $\mathbb{H}$, we also say that $\mathcal{H}$ is a largest hierarchy among the hierarchies of $\mathbb{H}$.

Guimarães et al. [2017] defined the following problem:

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

In order to solve this problem, Guimarães et al. [2017] formulated it as an optimization problem, where they were interested in finding a maximal not-too-coarse hierarchy for the HGB method.

The HGB method handles hierarchies by their weight maps and using the supremum and infimum between hierarchies. The relation "is larger/smaller than" on hierarchies can be characterized using the relation "is less/greater than" on weight maps. More formally, given two weight maps f and f_0 such that $f(u) \leq f_0(u)$ for any $u \in E$, the quasi-flat zone hierarchy of f (i.e., $\mathcal{QFZ}(T, f)$) is larger than the quasi-flat zone hierarchy of f_0 (i.e., $\mathcal{QFZ}(T, f_0)$). Indeed, this refers to the transformation of hierarchies that we present in Section 3.5.

Guimarães et al. [2017] proposed the HGB method to search the largest not-too-

coarse hierarchy by iteratively lowering the weights of a map. It initiates with a weight map where the weight of every edge is set to a maximum value (such initial weight map corresponds to the smallest not-too-coarse hierarchy). Then, in each iteration, the weight map is lowered. The final weight map corresponds to a large not-too-coarse hierarchy.

3.5 Transformation of hierarchies

Guimarães et al. [2017] proposed HGB as a method to hierarchize any image segmentation algorithm that works under the same machinery as the one proposed by Felzenszwalb and Huttenlocher [2004]. The HGB method relies on the tools provided by mathematical morphology (MM). Indeed, MM provides a set of tools and operators to modify hierarchies between different representations [Cousty et al., 2009, Xu et al., 2016]. Nonetheless, these approaches are not able to handle region merging predicates as the one used by Felzenszwalb and Huttenlocher [2004], Nock and Nielsen [2004], Haxhimusa et al. [2006]. Then, it was necessary to propose a method that deals with such merging predicates to produce a hierarchical segmentation.

In order to obtain a hierarchical segmentation, we rely on the framework proposed by Cousty et al. [2018] (see Section 2.3); the HGB method represent hierarchies by the edge weight maps and transformation of hierarchies. Guimarães et al. [2017] reformulated the region merging predicate of Felzenszwalb and Huttenlocher [2004] to produce a hierarchy. In the set of all possible hierarchies, Guimarães et al. [2017] proposed to find the "best possible" hierarchy as an optimization problem. In the following sections, we explain in more detail each of these points.

The HGB method is a graph-based image segmentation method. From the notions we presented in Chapter 2, an image is represented as an edge-weighted graph (G, w) from which we can compute its quasi-flat zones hierarchy $\mathcal{QFZ}(G, w)$. Let T be the minimum spanning tree of (G, w). We know from [Cousty et al., 2018] that the two hierarchies $\mathcal{QFZ}(G, w)$ and $\mathcal{QFZ}(T, w)$ are the same; then, it is possible to only work on the hierarchy $\mathcal{QFZ}(T, w)$.

The core idea of the HGB method relies on transforming the initial hierarchy $\mathcal{QFZ}(T, w)$ into a new hierarchy (see Section 2.3 of Chapter 2). In this transformation process, the hierarchical structure of the initial hierarchy is reconstructed according to a dissimilarity measure D. Since a hierarchy can be handled by its weight map, this transformation process is performed by producing a new weight map f. We will see in the next section, that the process only needs to deal with edges in T to produce f. The final result from this transformation is the hierarchy $\mathcal{QFZ}(T, f)$.

From the above explanation, we can see that the main result from this transformation step is the generation of the new weight function f for T. The generation of the weight map f resulted from using a dissimilarity measure D. One of the contributions from Guimarães et al. [2017] is that the HGB method is independent from which dissimilarity measure D we can use. As a case of study, Guimarães et al. [2017] used the dissimilarity measure originally proposed by Felzenszwalb and Huttenlocher [2004], but the HGB method itself can use other types of dissimilarities (this will be further studied in Chapter 6).

3.6 Definition of HGB method

The HGB method is presented in Method 1 as the hierarchization of segmentation methods that relies on a dissimilarity measure between two regions, *e.g.*, the one of the GB method proposed by Felzenszwalb and Huttenlocher [2004]. The core of such a method relies on the framework proposed by Cousty et al. [2018]; we presented in Chapter 2 about the bijection between hierarchies, quasi-flat zones, and saliency maps. In particular, the HGB method takes the graph representation of an image, then uses this initial weight map and a dissimilarity measure to compute a final quasi-flat zone hierarchy. This is possible thanks to the framework proposed by Cousty et al. [2018] where we can handle hierarchies through their weight maps and then transform them into other hierarchical representations; in this case, a quasi-flat zone hierarchy.

The HGB method takes as input an image represented by a graph G with its associated weight function w, where the minimum spanning tree T of G is taken indeed. From (T, w), HGB method computes a new weight function f which leads to a new hierarchy $\mathcal{H} = \mathcal{QFZ}(T, f)$. The resulting hierarchy $\mathcal{H}$ is considered as the hierarchical image segmentation of the initial image. Thus, the core of the method is the generation of the weight function f for T.

To compute the new map f, the HGB method first initializes all values of f to infinity (see Line 1). Then, an observation scale value $f(u)$ is computed for each edge $u \in E(T)$ in non-decreasing order with respect to the original weight w (see Line 2). Note that each iteration in the loop requires computing the hierarchy $\mathcal{H} = \mathcal{QFZ}(T, f)$ (see Line 3). Once $\mathcal{H}$ is obtained, the value $\lambda^\star_{\mathcal{H}}(u)$ of a finite subset $\mathbb{E}$ of $\mathbb{R}$ is obtained by the minimization:

$$\lambda^\star_{\mathcal{H}}(\{x, y\}) = \min \left\{ \lambda \in \mathbb{E} \mid D\left(\mathcal{H}^\lambda_x, \mathcal{H}^\lambda_y\right) \leq \lambda \right\}. \tag{3.3}$$

We first consider the regions $\mathcal{H}^\lambda_x$ and $\mathcal{H}^\lambda_y$ at a level λ. Using the dissimilarity measure

Method 1: HGB method

Input : A minimum spanning tree T of an edge-weighted graph (G, w)
Output: A hierarchy $\mathcal{H} = \mathcal{QFZ}(T, f)$

1 **for** *each* $u \in E(T)$ **do** $f(u) := \max\{\lambda \in \mathbb{E}\}$;
2 **for** *each* $u \in E(T)$ *in non-decreasing order for* w **do**
3 $\quad \mathcal{H} := \mathcal{QFZ}(T, f)$;
4 $\quad f(u) := \mathrm{p}_{\mathbb{E}}\left(\lambda^{\star}_{\mathcal{H}}(u)\right)$;
5 **end**
6 $\mathcal{H} := \mathcal{QFZ}(T, f)$;

D, we check if $D\left(\mathcal{H}^{\lambda}_{x}, \mathcal{H}^{\lambda}_{y}\right) \leq \lambda$. Equation (3.3) states that $\lambda^{\star}_{\mathcal{H}}(\{x, y\})$ is the minimum value λ for which this assertion holds true. Observe that the minimization involved in Equation (3.3) has a solution only if the maximum of $\mathbb{E}$ is greater than the maximum possible dissimilarity value. In the following, we assume that this assumption always holds true.

It should be mentioned that the non-hierarchical method proposed by Felzenszwalb and Huttenlocher [2004] guarantees to obtain partitions that are neither too fine nor too coarse for a given scale. On the other hand, the hierarchical method cannot provide simultaneously both of similar properties that are extended to hierarchies [Guimarães et al., 2017]. However, the later method allows us to find maximal not-too-coarse hierarchies.

As mentioned above, Guimarães et al. [2017] provided a general scheme to compute Method 1. In the next chapter, we discuss how the direct implementation of this scheme is not very efficient. In order to tackle this situation, we acknowledge the problem is twofold. Indeed, it is necessary to propose efficient (i.e., exact and fast) algorithms for (i) solving the minimization involved in Equation (3.3); and (ii) computing the quasi-flat zone hierarchy $\mathcal{QFZ}(T, f)$ at each iteration of Method 1 (Lines 3 and 6). In this thesis we study such problems and we propose exact solutions to both problems in Chapter 4.

3.7 Illustration of HGB method on a toy example

Figure 3.2 llustrates an example of application of Method 1, where (a) shows the input graph (T, w). The HGB scheme comprehends two parts, an initialization and the transformation of a hierarchy, they are performed in the following manner:

Initialization In this step, the weight map f is initialized with all values set to 9, which is the maximum value of $\mathbb{E}$. We denote this instance of the weight map f as f_0. In

Figure 3.2, (a) shows a small input graph, (b) shows the resulting weight map from the initialization step.

(a) Input graph (b) Initialization

Figure 3.2: Illustration of Method 1 with $\mathbb{E} = \{0, 1, \ldots, 9\}$: (a) the input graph (T, w). Images done by the author.

Transformation of hierarchy The hierarchy of graph (T, f) is transformed in each iteration of the method. This iteration begins in Line 2 of Method 1, where we process every edge $u = \{x, y\} \in E(T)$ in non-decreasing order. Then, for each iteration, we process Lines 3 and 4 in Method 1:

- Iteration 1, $u = \{a, b\}$:
  Line 3: We compute the quasi-flat zone hierarchy with the current weight map of f, that is $\mathcal{QFZ}(T, f_0)$ (see Figure 3.3(a)).
  Line 4: we obtain $\lambda^\star_{\mathcal{H}}(\{a, b\})$ by solving the minimization problem from Equation 3.3. In this initial setting we only have singletons on the hierarchy; thus, the minimum value $\lambda \in \mathbb{E}$ for which $D\left(\mathcal{H}^\lambda_a, \mathcal{H}^\lambda_b\right) \leq \lambda$ holds true is $\lambda = 2$, and $\mathrm{p}_{\mathbb{E}}\left(\lambda^\star_{\mathcal{H}}(\{a, b\})\right) = 1$. We set $f(\{a, b\}) = 1$ accordingly. This instance of the weight map f is denoted as f_1. Figure 3.3(b) shows the updated graph (T, f_1).

(a) $\mathcal{QFZ}(T, f_0)$ (b) (T, f_1)

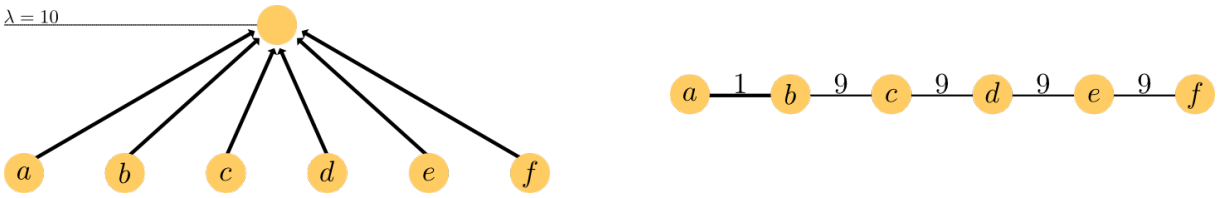


Figure 3.3: Illustration of iteration 1 of Method 1 for the input graph in Figure 3.2 . Images done by the author.

- Iteration 2, $u = \{c, d\}$:
  Line 3: We compute the quasi-flat zone hierarchy with the current weight map of f, that is $\mathcal{QFZ}(T, f_1)$ (see Figure 3.4(a)).
  Line 4: we obtain $\lambda^\star_{\mathcal{H}}(\{c, d\})$ by solving the minimization problem from Equation 3.3. In this initial setting we only have singletons on the hierarchy; thus, the minimum value $\lambda \in \mathbb{E}$ for which $D\left(\mathcal{H}^\lambda_c, \mathcal{H}^\lambda_d\right) \leq \lambda$ holds true is $\lambda = 2$, and $\mathrm{p}_{\mathbb{E}}\left(\lambda^\star_{\mathcal{H}}(\{c, d\})\right) = 1$.

We set $f(\{c,d\}) = 1$ accordingly. This instance of the weight map f is denoted as f_2. Figure 3.4(b) shows the updated graph (T, f_2).

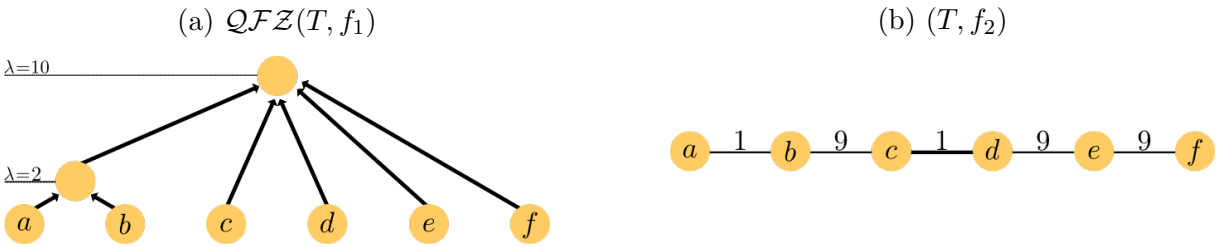


Figure 3.4: Illustration of iteration 2 of Method 1 for the input graph in Figure 3.2 . Images done by the author.

- Iteration 3, $u = \{b, c\}$:
  Line 3: We compute the quasi-flat zone hierarchy with the current weight map of f, that is $\mathcal{QFZ}(T, f_2)$ (see Figure 3.5(a)).
  Line 4: we obtain $\lambda^\star_{\mathcal{H}}(\{b,c\})$ by solving the minimization problem from Equation 3.3.

  We can see that for hierarchy $\mathcal{QFZ}(T, f_2)$, when $\lambda \in \{0, 1\}$, $\mathcal{H}^\lambda_b, \mathcal{H}^\lambda_c$ are singletons and $D\left(\mathcal{H}^\lambda_b, \mathcal{H}^\lambda_c\right) = 4$, and the condition $D\left(\mathcal{H}^\lambda_b, \mathcal{H}^\lambda_c\right) \leq \lambda$ is false. When $\lambda \in \{2, 3\}$, the regions $\mathcal{H}^\lambda_b$, and $\mathcal{H}^\lambda_c$ are not singletons. The inter- and intra- component values are as follows: $\Delta_{inter}(\mathcal{H}^\lambda_b, \mathcal{H}^\lambda_c) = 4$, $\Delta_{intra}(\mathcal{H}^\lambda_b) = 2$, and the area of region $|\mathcal{H}^\lambda_b| = 2$. Correspondingly, $\Delta_{inter}(\mathcal{H}^\lambda_c, \mathcal{H}^\lambda_b) = 4$, $\Delta_{intra}(\mathcal{H}^\lambda_c) = 2$, and the area of region $|\mathcal{H}^\lambda_c| = 2$, the dissimilarity measure is $D\left(\mathcal{H}^\lambda_b, \mathcal{H}^\lambda_c\right) = 4$, and the condition $D\left(\mathcal{H}^\lambda_b, \mathcal{H}^\lambda_c\right) \leq \lambda$ is also false.

  The minimum $\lambda \in \mathbb{E}$ for which $D\left(\mathcal{H}^\lambda_b, \mathcal{H}^\lambda_c\right) \leq \lambda$ holds true is $\lambda = 4$, and $\mathrm{p}_{\mathbb{E}}\left(\lambda^\star_{\mathcal{H}}(\{b,c\})\right) = 3$. We set $f(\{c,d\}) = 3$ accordingly. This instance of the weight map f is denoted as f_3. Figure 3.5(b) shows the updated graph (T, f_3).

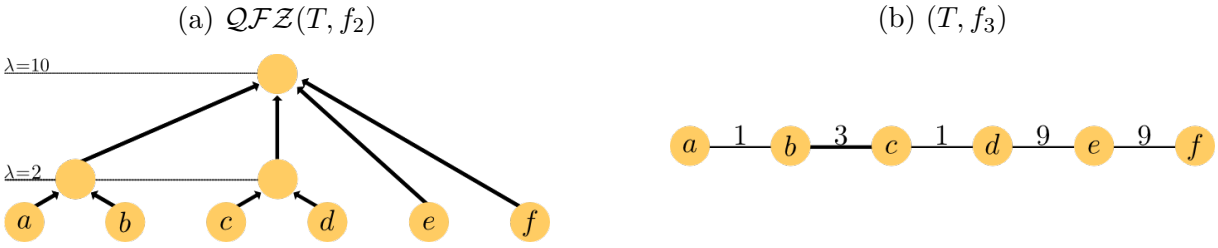


Figure 3.5: Illustration of iteration 3 of Method 1 for the input graph in Figure 3.2 . Images done by the author.

- Iteration 4, $u = \{e, f\}$:
  Line 3: We compute the quasi-flat zone hierarchy with the current weight map of f, that is $\mathcal{QFZ}(T, f_3)$ (see Figure 3.6(a)).

Line 4: we obtain $\lambda^{\star}_{\mathcal{H}}(\{e, f\})$ by solving the minimization problem from Equation 3.3.

We can see that for hierarchy $\mathcal{QFZ}(T, f_3)$, when $\lambda \in \{0, 1, 2, 3\}$, $\mathcal{H}^{\lambda}_e, \mathcal{H}^{\lambda}_f$ are singletons and $D\left(\mathcal{H}^{\lambda}_e, \mathcal{H}^{\lambda}_f\right) = 4$, and the condition $D\left(\mathcal{H}^{\lambda}_e, \mathcal{H}^{\lambda}_f\right) \leq \lambda$ is false. The minimum $\lambda \in \mathbb{E}$ for which $D\left(\mathcal{H}^{\lambda}_e, \mathcal{H}^{\lambda}_f\right) \leq \lambda$ holds true is $\lambda = 4$, and $\mathrm{p}_{\mathbb{E}}\left(\lambda^{\star}_{\mathcal{H}}(\{e, f\})\right) = 3$. We set $f(\{e, f\}) = 3$ accordingly. This instance of the weight map f is denoted as f_4. Figure 3.6(b) shows the updated graph (T, f_4).

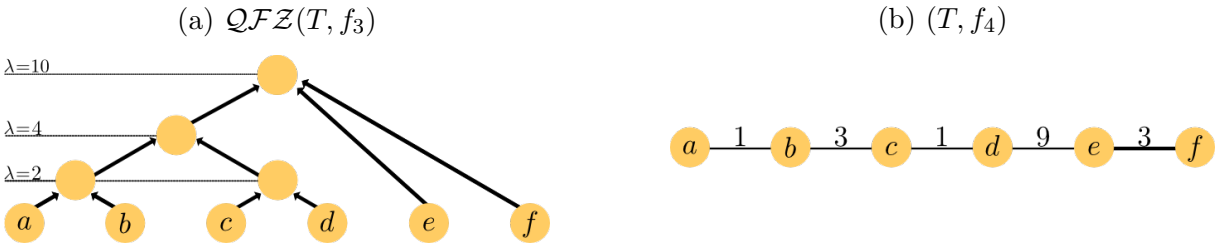


Figure 3.6: Illustration of iteration 4 of Method 1 for the input graph in Figure 3.2 . Images done by the author.

- Iteration 5, $u = \{d, e\}$:
  Line 3: We compute the quasi-flat zone hierarchy with the current weight map of f, that is $\mathcal{QFZ}(T, f_4)$ (see Figure 3.7(a)).
  Line 4: we obtain $\lambda^{\star}_{\mathcal{H}}(\{d, e\})$ by solving the minimization problem from Equation 3.3.

  We can see that for hierarchy $\mathcal{QFZ}(T, f_4)$; when $\lambda \in \{0, 1\}$, $\mathcal{H}^{\lambda}_d, \mathcal{H}^{\lambda}_e$ are singletons and $D\left(\mathcal{H}^{\lambda}_d, \mathcal{H}^{\lambda}_e\right) = 5$, and the condition $D\left(\mathcal{H}^{\lambda}_d, \mathcal{H}^{\lambda}_e\right) \leq \lambda$ is false. When $\lambda \in \{2, 3\}$, the region $\mathcal{H}^{\lambda}_d$ is not a singleton. The inter- and intra- component values are as follows: $\Delta_{inter}(\mathcal{H}^{\lambda}_d, \mathcal{H}^{\lambda}_e) = 5$, $\Delta_{intra}(\mathcal{H}^{\lambda}_d) = 2$, and the area of region $|\mathcal{H}^{\lambda}_d| = 2$. Region $\mathcal{H}^{\lambda}_e$ is a singleton, and the dissimilarity measure $D\left(\mathcal{H}^{\lambda}_d, \mathcal{H}^{\lambda}_e\right) = 6$, thus the condition $D\left(\mathcal{H}^{\lambda}_d, \mathcal{H}^{\lambda}_e\right) \leq \lambda$ is also false.

  The minimum $\lambda \in \mathbb{E}$ for which $D\left(\mathcal{H}^{\lambda}_d, \mathcal{H}^{\lambda}_e\right) \leq \lambda$ holds true is $\lambda = 4$, and $\mathrm{p}_{\mathbb{E}}\left(\lambda^{\star}_{\mathcal{H}}(\{d, e\})\right) = 3$. We set $f(\{d, e\}) = 3$ accordingly. This instance of the weight map f is denoted as f_5. Figure 3.7(b) shows the updated graph (T, f_5).

After iteration 5, all the edges of the graph (T, w) have been processed and the weight map f is totally updated. We show the resulting hierarchy $\mathcal{QFZ}(T, f)$ in Figure 3.7(c). In Figure 3.8, we also illustrate segmentation results obtained from the original HGB method.

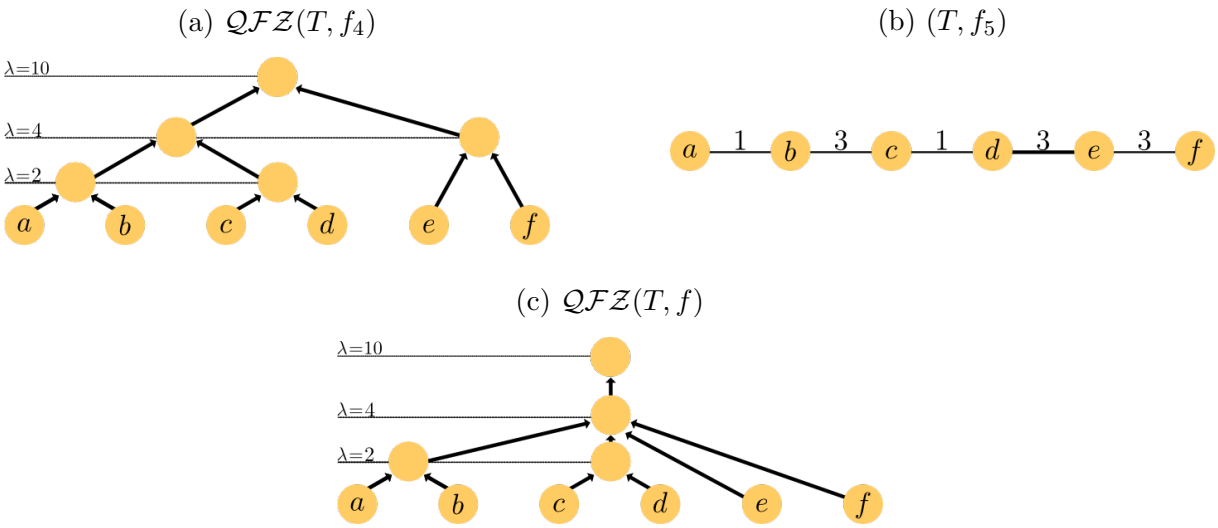


Figure 3.7: Illustration of iteration 5 of Method 1 for the input graph in Figure 3.2. (c) shows the resulting hierarchy of the HGB method. Images done by the author.

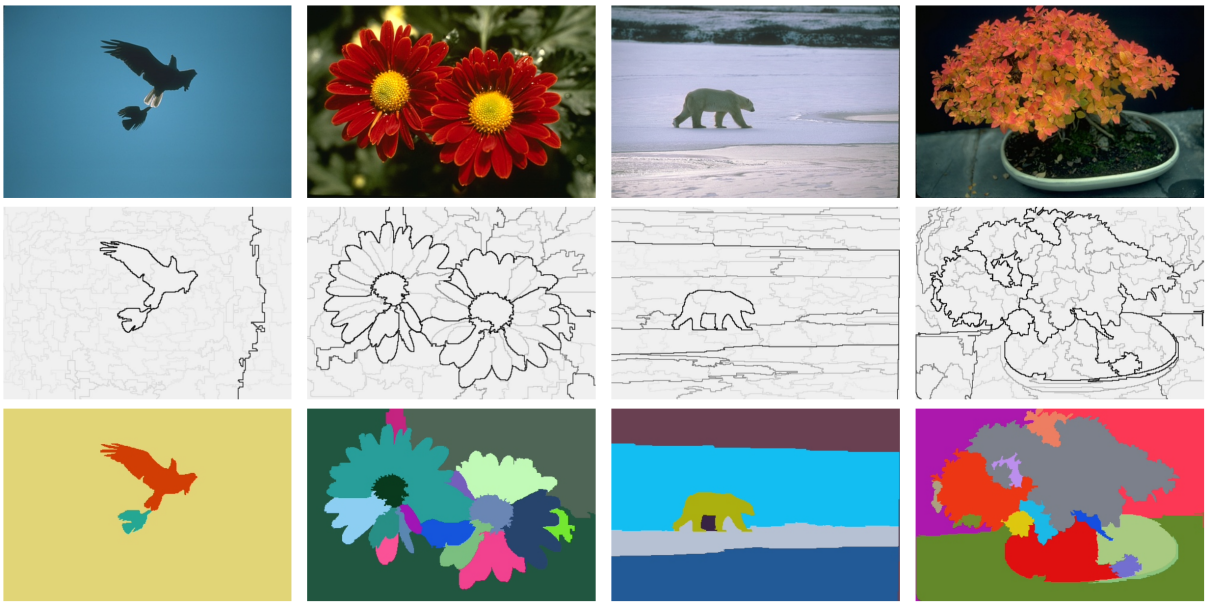


Figure 3.8: Illustration of segmentation results computed from the HGB method. The first row shows some input images, the middle row shows the hierarchy computed by the HGB method (using its saliency map representation), and the bottom row shows an image segmentation extracted from each corresponding hierarchy. (Image extracted from [Guimarães et al., 2017]).

3.8 Concluding remarks

In this chapter, we reviewed the hierarchical image segmentation method HGB. The importance of this chapter is to understand the machinery behind the HGB method. It is proposed as a method to hierarchize any image segmentation algorithm that relies on a region merging predicate. To this end, HGB relies on the notion of transformation

of hierarchies and the region merging predicate of the graph-based algorithm proposed by Felzenszwalb and Huttenlocher [2004]. This region merging predicate was reformulated as an observation scale dissimilarity, which allows the method to compute the scale at which two adjacent regions merge.

This initial understanding of the HGB method allowed us to recognize its properties and propose a formal definition of this method. It also helped us to formally extend the well-established GB method to a hierarchical image segmentation approach, which is one of the objectives of this thesis. During this chapter, we also pointed out and discussed the parts where we have acknowledged computational problems and research opportunities; we propose their solutions in the next chapters of this thesis. In particular, we acknowledged the lack of efficient algorithms for the computation of this method; we tackle this problem in Chapter 4, where we propose a new theoretical background that precisely formalizes the properties of the HGB method and allows the hierarchization of the GB method. We also acknowledged that the resulting hierarchy from the HGB method is based on the selection of an observation scale value. We explore new techniques for the selection of other interesting observation scale values in Chapter 5. Another research opportunity we acknowledge is the exploration of new regional attributes that can be computed for each region and that can capture other types of image characteristics.

Chapter 4

Efficient algorithms for hierarchical image graph-based segmentation

4.1 Introduction

In Chapter 3, we presented the HGB method and we acknowledged that one shortcoming of this work is that a practical algorithm to compute efficiently the results of the HGB method is not provided by Guimarães et al. [2017]. In this chapter, we present the elaboration of a precise formalism to study the properties of the HGB method, introduced in Chapter 3, and that allowed the hierarchization of the GB method. The results of this new theoretical background allow us to present in this chapter a series of exact and efficient algorithms for the computation of the HGB method, which are the first main contributions of this thesis.

From our previous studies, we note that the core of the HGB method is based on solving a minimization problem whose solution is the minimum observation scale at which adjacent regions in the image have to be merged (see Eq. (3.3)). Therefore, in Section 4.2, we discuss how to solve such minimization problem. We first discuss a straightforward implementation of the HGB method, which leads to an initial naive algorithm. We find that the naive algorithm to compute the minimization step is not very efficient. Therefore, we present the notions to discretize and consequently reduce the search space of the minimization problem; thus reducing the number of computations needed to solve the minimization problem. By reducing the search space we are able to avoid redundant computations, and consequently we propose two new efficient algorithms to compute the solution.

Also, from our previous studies, we see that the HGB method performs an update of the hierarchy at each of its iterations. Indeed, we can see that in Method 1, the weight map is updated in each of its iterations. Since the weight map is modified, the hierarchy itself is also modified. In Section 4.3, we study this procedure and we discuss its straightforward implementation which involves computing a new hierarchy in each

iteration with the current weight map. One approach can be using an efficient algorithm, such as the one proposed by Najman et al. [2013], to compute it at every iteration of the HGB method. Nonetheless, this creates the overhead of computing a quasi-flat zone hierarchy in every iteration. In order to overcome this situation, we discuss how to exploit the tree-based hierarchical representation of a quasi-flat zone hierarchy, in order to perform incremental updates of the hierarchy. From this study, we propose a new algorithm that only updates at each iteration the existing quasi-flat zone hierarchy instead of recomputing it from scratch. This is done with a procedure similar to the one proposed in [Havel et al., 2019, Wilkinson et al., 2008].

All of our new algorithms are validated in Section 4.4, where we present the experiment we performed to corroborate the efficiency of our algorithms. Finally, in Section 4.5, we present some concluding remarks about the theoretical and algorithmic contributions we made on this chapter, and how they relate to the objectives of this thesis. This chapter consists of the work presented in [Cayllahua-Cahuina et al., 2019b, Cayllahua Cahuina et al., 2018].

4.2 Solving HGB minimization problem

4.2.1 A naive algorithm for HGB method

Method 1 in Chapter 3 presents the HGB method; as inputs the method receives the minimum spanning tree T of a graph $G = (V, E)$ with its associated weight map w. The straightforward algorithm to implement such method would have to follow the next steps:

(i) An initialization of a weight map f to a maximum value in $\mathbb{E}$.

(ii) Compute for every edge u in $E(T)$, in non-decreasing order, the weight map value $f(u)$ as a result of the minimization problem defined in Equation (3.3).

(iii) The final weight map f is used to construct the final hierarchy $\mathcal{H}$ of the graph (T, f).

Step (i) and (iii) can be easily implemented; on the other hand, step (ii) involves two situations. The first involves solving the minimization problem presented in Equation (3.3) (see Line 4 in Method 1). The second is computing a quasi-flat zone hierarchy, in each iteration, for the current graph (T, f) (see Line 3 in Method 1).

4.2.1.1 Naive minimization algorithm

We first present a naive algorithm, namely Algorithm 1, to compute the value $\lambda^\star_{\mathcal{H}}(\{x, y\})$ given a hierarchy $\mathcal{H}$ and an edge $\{x, y\}$. According to Equation (3.3), it simply consists of considering the values of $\mathbb{E}$ in increasing order until finding a value $\lambda \in \mathbb{E}$ such that $D\left(\mathcal{H}^\lambda_x, \mathcal{H}^\lambda_y\right) \leq \lambda$. We remark that, when $\mathbb{E}$ is a set of consecutive integers, for any λ in $\mathbb{E}$, the result of $\mathrm{n}_{\mathbb{E}}(\lambda)$ and $\mathrm{p}_{\mathbb{E}}(\lambda)$ can be obtained with the simple integer instruction $\lambda + 1$ and $\lambda - 1$, respectively.

As said by Carlinet and Géraud [2014], the size and the maximal edge-value of each region of $\mathcal{H}$ can be computed on the fly at Line 3 of Method 1. Thus, once the regions $\mathcal{H}^\lambda_x$ and $\mathcal{H}^\lambda_y$ are identified the dissimilarity between them can be computed in constant time. Furthermore, if the hierarchy $\mathcal{H}$ is represented by its dendrogram, the regions $\mathcal{H}^\lambda_x$ and $\mathcal{H}^\lambda_y$ can be obtained at each iteration of the main loop of *Algorithm* 1 in constant time (more details about the implementation of such operation are provided in the following Section 4.2.4). Therefore, the time complexity of Algorithm 1 is $O(|\mathbb{E}|)$.

Algorithm 1: HGB Naive minimization of Equation (3.3)

Input : A hierarchy $\mathcal{H}$, an edge $\{x, y\}$
Output: The value $\lambda^\star$ such that $\lambda^\star = \lambda^\star_{\mathcal{H}}(\{x, y\})$

```
1 λ* := min{λ ∈ 𝔼} ;
2 while D(H^λ_x, H^λ_y) > λ* do
3 |   λ* := n_𝔼(λ*) ;
4 end
```

4.2.2 Stable intervals and stable partitions

In this section, we propose a general framework to solve efficiently the minimization problem presented in Equation (3.3). To this end, we establish that on certain subdomains, called stable intervals, the solution depends only on the bounds of the subdomain instead of all of its elements, reducing the computation complexity for the solution in such subdomain to a constant number of operations instead of a number of operations which depends linearly on the size of the subdomain (see Property 3). Then, in Theorem 5, we state that the problem on the whole domain can be solved by considering any partition of the domain into stable intervals and by solving the problem in each of these

stable intervals. This theorem is the fundamental result that allows us for proposing efficient minimization algorithms in Sections 4.2.3 and 4.2.4. More precisely, in the following sections, partitions of the domain in stable intervals that can be handled efficiently in a computerized procedure are presented.

Let λ_1 and λ_2 be any two real numbers in $\mathbb{E} \cup \{-\infty\}$ such that $\lambda_1 < \lambda_2$. We denote by $]\!]\lambda_1, \lambda_2]\!]_{\mathbb{E}}$ the subset of $\mathbb{E}$ that contains every element of $\mathbb{E}$ that is both greater than λ_1 and not greater than λ_2:

$$]\!]\lambda_1, \lambda_2]\!]_{\mathbb{E}} = \{\lambda \in \mathbb{E} \mid \lambda_1 < \lambda \leq \lambda_2\}. \tag{4.1}$$

We say that a subset I of $\mathbb{E}$ is an *open-closed interval of* $\mathbb{E}$, or simply an *interval*, if there exist two real values λ_1 and λ_2 such that I is equal to $]\!]\lambda_1, \lambda_2]\!]_{\mathbb{E}}$.

Definition 1 (stable interval)**.** *Let f be any map from $E(T)$ in $\mathbb{E}$, let $u = \{x, y\}$ be any edge in $E(T)$, and let $I =]\!]\lambda_1, \lambda_2]\!]_{\mathbb{E}}$ be any interval. We say that I is a* stable interval for (f, u) *if, for any two values λ and λ' in I, the two following statements hold true:*

1. *the regions containing x in the λ-level partition of (T, f) and in the λ'-level partition of (T, f) are equal; and*
2. *the regions containing y in the λ-level partition of (T, f) and in the λ'-level partition of (T, f) are equal.*

In other words, the interval $I =]\!]\lambda_1, \lambda_2]\!]_{\mathbb{E}}$ is a stable interval for (f, u) if, for any two values λ and λ' in I, we have $\mathcal{H}_x^{\lambda} = \mathcal{H}_x^{\lambda'}$ and $\mathcal{H}_y^{\lambda} = \mathcal{H}_y^{\lambda'}$, where $\mathcal{H}$ is the quasi-flat zone hierarchy of T for f. Hence, the following lemma can be straightforwardly deduced from the definition of the dissimilarity measure D.

Lemma 2. *Let f be any map from $E(T)$ in $\mathbb{E}$ and let $\mathcal{H}$ be the quasi-flat zone hierarchy of T for f, let $u = \{x, y\}$ be any edge in $E(T)$, and let $I =]\!]\lambda_1, \lambda_2]\!]_{\mathbb{E}}$ be a stable interval for (f, u). Then, for any λ in I, we have: $D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) = D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2})$.*

The following property firstly shows a necessary and sufficient condition on which the minimization problem presented in Equation (3.3) admits a solution when its domain of definition is restricted to a stable interval. Moreover, when the problem admits a solution over a certain stable interval, the following property provides an expression of this solution that depends only on the higher bound of the considered stable interval.

Property 3. *Let f be any map from $E(T)$ in $\mathbb{E}$, let $\mathcal{H}$ be the quasi-flat zone hierarchy of T for f, let $u = \{x, y\}$ be any edge in $E(T)$, and let $I =]\!]\lambda_1, \lambda_2]\!]_{\mathbb{E}}$ be a stable interval for (f, u). Then, the two following statements hold true:*

1. *the set $\{\lambda \in I \mid D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) \leq \lambda\}$ is nonempty if and only if $D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2})$ is not greater than λ_2 (*i.e.*, $D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2}) \leq \lambda_2$); and*

2. if $D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2}) \leq \lambda_2$, *then*

$$\min\left\{\lambda \in I \mid D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) \leq \lambda\right\} = \max\left\{n_{\mathbb{E}}\left(\lambda_1\right), \hat{n}_{\mathbb{E}}\left(D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2})\right)\right\}. \tag{4.2}$$

Proof. In order to prove Property 3, we consider three distinct cases. For each of these cases, we prove that $\{\lambda \in I \mid D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) \leq \lambda\}$ is either empty or nonempty and in the cases where it is nonempty, we establish that Equation (4.2) holds true. For each of these three cases, the associated proofs are graphically illustrated in Figure 4.1.

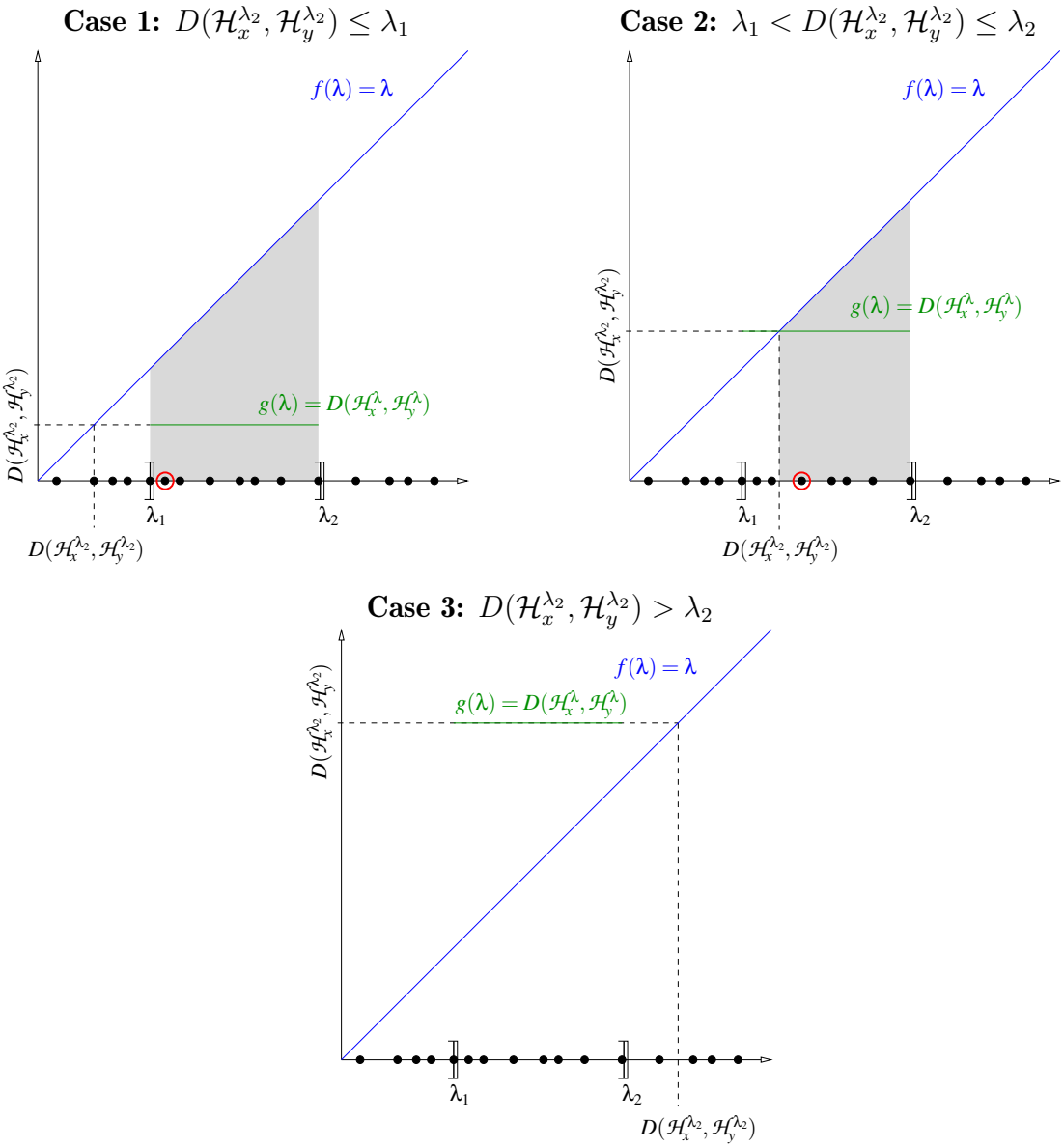


Figure 4.1: Graphical illustrations for the proof of Property 3. The elements of $\mathbb{E}$ are represented by dots on the horizontal axis of real values. The subdomain of E which contains every element λ of $I =]\!]\lambda_1, \lambda_2]\!]_{\mathbb{E}}$ such that $D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) \leq \lambda$ is represented by the gray zones and, when this subdomain is not empty, the unique solution to $\min\left\{\lambda \in I \mid D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) \leq \lambda\right\}$ is indicated by the circled dot.

1. Let us first assume that $D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2}) \leq \lambda_1$. By Lemma 2, we deduce that, for any $\lambda \in]\!]\lambda_1, \lambda_2]\!]_{\mathbb{E}}$, we have $D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) \leq \lambda_1$. Hence, for any λ in $I =]\!]\lambda_1, \lambda_2]\!]_{\mathbb{E}}$, we have $D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) \leq \lambda$. Since λ_2 belongs to I, in this case, the set $\{\lambda \in I \mid D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) \leq \lambda\}$ is nonempty and we deduce that $\min\{\lambda \in I \mid D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) \leq \lambda\}$ is the minimum of I, which is precisely $\mathrm{n}_{\mathbb{E}}(\lambda_1)$. Thus, we have $\min\{\lambda \in I \mid D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) \leq \lambda\} = \mathrm{n}_{\mathbb{E}}(\lambda_1)$. Furthermore, since $D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2}) \leq \lambda_1$, we have $\hat{\mathrm{n}}_{\mathbb{E}}\left(D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2})\right) \leq \hat{\mathrm{n}}_{\mathbb{E}}(\lambda_1)$. Therefore, since λ belongs to $\mathbb{E}$, we deduce that $\hat{\mathrm{n}}_{\mathbb{E}}\left(D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2})\right) < \mathrm{n}_{\mathbb{E}}(\lambda_1)$. Hence, in the considered case, we have $\min\left\{\lambda \in I \mid D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) \leq \lambda\right\} = \max\left\{\mathrm{n}_{\mathbb{E}}(\lambda_1), \hat{\mathrm{n}}_{\mathbb{E}}\left(D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2})\right)\right\}$.

2. Let us now assume that $\lambda_1 < D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2}) \leq \lambda_2$. Since λ_2 belongs to $I =]\!]\lambda_1, \lambda_2]\!]_{\mathbb{E}}$, we deduce that, in this case, the set $\{\lambda \in I \mid D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) \leq \lambda\}$ is nonempty. By Lemma 2, for any $\lambda \in I$, we have $D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) = D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2})$. Then, for any λ in I such that $\lambda \geq D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2})$, we have $D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) \leq \lambda$ and, for any λ in I such that $\lambda < D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2})$, we have $D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) > \lambda$. Hence, in this case, we deduce that $\min\{\lambda \in I \mid D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) \leq \lambda\}$ is the minimum value of $\mathbb{E}$ which is not less than $D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2})$. Thus, in this case, we have $\min\{\lambda \in I \mid D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) \geq \lambda\} = \hat{\mathrm{n}}_{\mathbb{E}}\left(D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2})\right)$. Furthermore, since $\lambda_1 < D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2})$, we deduce that $\mathrm{n}_{\mathbb{E}}(\lambda_1) \leq \hat{\mathrm{n}}_{\mathbb{E}}\left(D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2})\right)$. Thus, in this case, we also have $\min\{\lambda \in I \mid D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda})\} = \max\left\{\mathrm{n}_{\mathbb{E}}(\lambda_1), \hat{\mathrm{n}}_{\mathbb{E}}\left(D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2})\right)\right\}$.

3. Let us finally assume that $\lambda_2 < D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2})$. By Lemma 2, for any $\lambda \in I$, we have $D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) = D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2})$. Hence, for any λ in I, we have $D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) > \lambda$. Hence, in this case the set $\{\lambda \in I \mid D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) \leq \lambda\}$ is empty.

From the statements given in cases (1), (2), and (3) above, we can affirm that $\{\lambda \in I \mid D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) \leq \lambda\}$ is nonempty if and only if $D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2}) \leq \lambda_2$, which completes the proof of Property 3.1. Furthermore, from the statements given in cases (1) and (2), we deduce that, if $D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2}) \leq \lambda_2$, then we have $\min\left\{\lambda \in I \mid D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda}) \leq \lambda\right\} = \max\left\{\mathrm{n}_{\mathbb{E}}(\lambda_1), \hat{\mathrm{n}}_{\mathbb{E}}\left(D(\mathcal{H}_x^{\lambda_2}, \mathcal{H}_y^{\lambda_2})\right)\right\}$, which completes the proof of Property 3.2. □

Definition 4 (stable partition)**.** *Let f be any map from $E(T)$ in $\mathbb{E}$, let $u = \{x, y\}$ be any edge in $E(T)$. Let $\mathbb{I} = (\lambda_0, \ldots, \lambda_\ell)$ be a series of real values in $\mathbb{E} \cup \{-\infty\}$ such that $\lambda_0 = -\infty$, $\lambda_\ell = \max\{\lambda \in \mathbb{E}\}$, and, for any i in $\{1, \ldots, \ell\}$, we have $\lambda_i > \lambda_{i-1}$. Let $\mathcal{P}_{\mathbb{I}} = \{]\!]\lambda_0, \lambda_1]\!]_{\mathbb{E}}, \ldots,]\!]\lambda_{\ell-1}, \lambda_\ell]\!]_{\mathbb{E}}\}$. The series $\mathbb{I}$ (resp. the set $\mathcal{P}_{\mathbb{I}}$) is called a* stable bound series (resp. a stable partition) of $\mathbb{E}$ for (f, u) *if any element of $\mathcal{P}_{\mathbb{I}}$ is a stable interval for (f, u).*

Given a stable bound series $\mathbb{I} = (\lambda_0, \ldots, \lambda_\ell)$, it can be easily seen that the stable partition $\mathcal{P}_{\mathbb{I}}$ is a partition of $\mathbb{E}$. Moreover, given the stable bound series $\mathbb{I} = (\lambda_0, \ldots, \lambda_\ell)$, we denote by $\mathrm{ind}(\mathbb{I})$ the set of indices $\{1, \ldots, \ell\}$.

The following theorem, which is the main result of this section, shows that given a stable bound series, the minimization problem defined by Equation (3.3) can be solved by considering only the elements of this series rather than all elements of the domain $\mathbb{E}$. Hence, this result is a keystone to provide an efficient algorithm for computing $\lambda^{\star}_{\mathcal{H}}(\{x,y\})$ with a reduced complexity compared to the naive algorithm (Algorithm 1).

Theorem 5. *Let f be any map from E in $\mathbb{E}$, let $\mathcal{H}$ be the quasi-flat zone hierarchy of G for f, let $u = \{x,y\}$ be any edge in E, and let $\mathbb{I} = (\lambda_0, \ldots, \lambda_\ell)$ be a stable bound series of $\mathbb{E}$ for (f,u). Then, the following statement holds true:*

$$\lambda^{\star}_{\mathcal{H}}(\{x,y\}) = \min\{\max\{n_{\mathbb{E}}(\lambda_{i-1}), \hat{n}_{\mathbb{E}}\left(D(\mathcal{H}^{\lambda_i}_x, \mathcal{H}^{\lambda_i}_y)\right)\} \mid i \in ind(\mathbb{I}), D(\mathcal{H}^{\lambda_i}_x, \mathcal{H}^{\lambda_i}_y) \leq \lambda_i\}. \quad (4.3)$$

Proof. Let $I_i =]\!]\lambda_{i-1}, \lambda_i]\!]_{\mathbb{E}}$, for any i in $\mathrm{ind}(\mathbb{I}) = \{1, \ldots, \ell\}$. It can be easily seen that $\mathcal{P}_{\mathbb{I}} = \{I_1, \ldots, I_\ell\}$. Since $\mathcal{P}_{\mathbb{I}}$ is a partition of $\mathbb{E}$, we may affirm that:

$$\min\left\{\lambda \in \mathbb{E} \mid D(\mathcal{H}^{\lambda}_x, \mathcal{H}^{\lambda}_y) \leq \lambda\right\} = \min\left\{\min\left\{\lambda \in I_i \mid D(\mathcal{H}^{\lambda}_x, \mathcal{H}^{\lambda}_y) \leq \lambda\right\} \mid i \in \mathrm{ind}(\mathbb{I}), \left\{\lambda \in I_i \mid D(\mathcal{H}^{\lambda}_x, \mathcal{H}^{\lambda}_y) \leq \lambda\right\} \neq \emptyset\right\}$$

Since, for any $i \in \{1, \ldots, \ell\}$ the interval $I_i =]\!]\lambda_{i-1}, \lambda_i]\!]_{\mathbb{E}}$ is a stable interval, by Property 1, we deduce that:

$$\min\left\{\lambda \in \mathbb{E} \mid D(\mathcal{H}^{\lambda}_x, \mathcal{H}^{\lambda}_y) \leq \lambda\right\} = \min\{\max\{\mathrm{n}_{\mathbb{E}}(\lambda_{i-1}), \hat{\mathrm{n}}_{\mathbb{E}}\left(D(\mathcal{H}^{\lambda_i}_x, \mathcal{H}^{\lambda_i}_y)\right)\} \mid i \in \mathrm{ind}(\mathbb{I}), D(\mathcal{H}^{\lambda_i}_x, \mathcal{H}^{\lambda_i}_y) \leq \lambda_i\}$$

□

4.2.3 Minimization by range

Let f be any map from E to $\mathbb{E}$, the *range of* f, denoted by $\mathrm{range}(f)$, is the set of values that f can take as its argument varies over E: $\mathrm{range}(f) = \{f(u) \mid u \in E\}$. We denote by $\mathbb{R}_f$ the series $(\lambda_0, \ldots, \lambda_\ell)$ of ordered values in $\mathrm{range}(f) \cup \{-\infty, \max\{\lambda \in \mathbb{E}\}\}$:

1. $\lambda_0 = -\infty$, and $\lambda_\ell = \max\{\lambda \in \mathbb{E}\}$;
2. $\{\lambda_i \mid i \in \{1, \ldots, \ell - 1\}\} = \mathrm{range}(f) \setminus \{\max\{\lambda \in \mathbb{E}\}\}$; and
3. for any i in $\{1, \ldots \ell\}$, we have $\lambda_i > \lambda_{i-1}$.

Property 6. *Let f be any map from $E(T)$ in $\mathbb{E}$ and let $u = \{x, y\}$ be any edge in $E(T)$. Then, the series $\mathbb{R}_f$ is a stable bound series for (f, u).*

Proof. Since $\lambda_0 = -\infty$, since $\lambda_\ell = \max\{\lambda \in \mathbb{E}\}$, and since, for any $i \in \{1, \dots \ell\}$ we have $\lambda_i < \lambda_{i-1}$, in order to establish Property 6, it is sufficient to prove that, for any i in $\{1, \dots, \ell\}$, the interval $]\!]\lambda_{i-1}, \lambda_i]\!]_{\mathbb{E}}$ is a stable interval for (f, u). Let i be any element in $\{1, \dots, \ell\}$. Let λ be any element in $]\!]\lambda_{i-1}, \lambda_i]\!]_{\mathbb{E}}$. Let v be any edge of T which belongs to $f_{\lambda_i}(T)$. By Equation (2.1), we have $f(v) < \lambda_i$. By definition of $\mathbb{R}_f$, there exists $j \in \{1, \dots, i-1\}$ such that $f(v) = \lambda_j$. Thus, we have $f(v) < \lambda$, which implies, by Equation (2.1), that v also belongs to $f_{\lambda_i}(T)$. Furthermore, since $\lambda \leq \lambda_i$, for any edge e such that $f(e) \geq \lambda_i$ we also have $f(e) \geq \lambda$. Hence, by Equation (2.1), any edge which does not belong to $f_{\lambda_i}(T)$ does not belong to $f_\lambda(T)$ either. Therefore, we deduce that $w_\lambda(T) = w_{\lambda_i}(T)$. Thus, since $\mathcal{H}$ is the quasi-flat zone hierarchy of f, we also have $\mathcal{H}_x^\lambda = \mathcal{H}_x^{\lambda_i}$ and $\mathcal{H}_y^\lambda = \mathcal{H}_y^{\lambda_i}$, by definition of a quasi-flat zone hierarchy. Hence, the interval $]\!]\lambda_{i-1}, \lambda_i]\!]_{\mathbb{E}}$ is a stable interval for (f, u). □

Corollary 7. *Let f be any map from $E(T)$ in $\mathbb{E}$, let $\mathcal{H}$ be the quasi-flat zone hierarchy of T for f, let $u = \{x, y\}$ be any edge in $E(T)$, and let $\mathbb{R}_f = (\lambda_0, \dots, \lambda_\ell)$. Then, the following statement holds true:*

$$\lambda^\star_{\mathcal{H}}(\{x, y\}) = \min\{\max\{n_{\mathbb{E}}(\lambda_{i-1}), \hat{n}_{\mathbb{E}}\left(D(\mathcal{H}_x^{\lambda_i}, \mathcal{H}_y^{\lambda_i})\right)\} \mid i \in ind(\mathbb{R}_f), D(\mathcal{H}_x^{\lambda_i}, \mathcal{H}_y^{\lambda_i}) \leq \lambda_i\}. \quad (4.4)$$

According to Corollary 7, we can compute $\lambda^\star_{\mathcal{H}}(\{x, y\})$ by browsing the values of $\mathbb{R}_f$ in increasing order until a value λ_i such that $D(\mathcal{H}_x^{\lambda_i}, \mathcal{H}_y^{\lambda_i}) \leq \lambda_i$ is found and by setting the value of $\lambda^\star_{\mathcal{H}}(\{x, y\})$ to the maximum between $\mathrm{n}_{\mathbb{E}}(\lambda_{i-1})$ and $\hat{\mathrm{n}}_{\mathbb{E}}\left(D(\mathcal{H}_x^{\lambda_i}, \mathcal{H}_y^{\lambda_i})\right)$. In order to make such process computable, it is necessary to browse the range of f in increasing order. To this end, we propose to store the values of f in a sorted linked list. Algorithm 2 provides a precise description of this process. It can be observed that when the value $\mathrm{p}_{\mathbb{E}}(\lambda^\star_{\mathcal{H}}(\{x, y\}))$ is not yet present in the range of f, the linked list representing this range is updated so that it is ready for the next iteration of the main loop in Method 1. It has to be also noted that in Method 1, the weight of every edge is initialized to the maximal value of $\mathbb{E}$. In other words, the linked list must be initialized in Method 1 with the singleton $\{\max\{\lambda \in \mathbb{E}\}\}$.

As said at the end of Section 4.2.1.1, at each iteration of the while loop in Algorithm 2, the dissimilarity between $\mathcal{H}_x^\lambda$ and $\mathcal{H}_y^\lambda$ can be obtained in constant time. The instructions and tests of the while loop in Algorithm 2 are executed at most $|\mathrm{range}(f)|+1$ times and each of these instructions can be made in constant time. Thus, Algorithm 2 runs in $O(|\mathrm{range}(f)|)$ time complexity. It has to be noted that $|\mathrm{range}(f)|$ is always less than the number of edges of T, which is equal to the number of vertices of T minus one, since T is a tree. This number is, in general, much less than the number of elements

Algorithm 2: HGB Minimization by range

Input : A hierarchy $\mathcal{H}$, a weight map f such that $\mathcal{H} = \mathcal{QFZ}(T, f)$, an edge $\{x, y\}$ of T, a linked list L of the values of $\mathbb{R}_f$ in increasing order

Output: The value $\lambda^\star$ such that $\lambda^\star = \lambda^\star_{\mathcal{H}}(\{x, y\})$, the updated linked list L of the values of $\mathbb{R}_f \cup \{\mathrm{p}_{\mathbb{E}}(\lambda^\star)\}$ in increasing order

$l := L.head$; $\lambda := l.value$; $\lambda_{prev} := -\infty$;
while $D\left(\mathcal{H}^\lambda_x, \mathcal{H}^\lambda_y\right) > \lambda$ **do**
  $\lambda_{prev} := \lambda$; $l := l.next$; $\lambda := l.value$;
end
$\lambda^\star := \max(\mathrm{n}_{\mathbb{E}}(\lambda_{prev}), \hat{\mathrm{n}}_{\mathbb{E}}\left(D(\mathcal{H}^\lambda_x, \mathcal{H}^\lambda_y)\right))$;
if $\mathrm{p}_{\mathbb{E}}(\lambda^\star) \neq \lambda_{prev}$ **then** $L.insert(\mathrm{p}_{\mathbb{E}}(\lambda^\star))$;

of $\mathbb{E}$. Indeed, for reaching a good precision, $\mathbb{E}$ can be chosen as the set of all possible values of the dissimilarity measure D. In such case, the number of elements in $\mathbb{E}$ is in the order of $|\mathrm{range}(w)| \times |V|$. Hence, in this case, the time-complexity is reduced from $O(|\mathrm{range}(w)| \times |V|)$ with Algorithm 1 to $O(|V|)$ with Algorithm 2.

4.2.4 Minimization by branch

In the previous section, we reduced the size of the search space of the minimization defined in Equation (3.3) by considering the range $\mathbb{R}_f$ of the function f (*i.e.*, a characteristic function of the considered hierarchy $\mathcal{H}$) instead of the set $\mathbb{E}$ of all possible scales of the hierarchy $\mathcal{H}$ (see Corollary 7). In this section, we show that this search space can be further reduced, leading to a third algorithm for computing the value $\lambda^\star_{\mathcal{H}}(\{x, y\})$, given any hierarchy $\mathcal{H}$ and any edge $\{x, y\}$.

In order to obtain this second reduction, we observe in Equation (3.3) that the only regions of the hierarchy involved in the minimization are those containing x and y. Therefore, while searching for the value $\lambda^\star_{\mathcal{H}}(\{x, y\})$, it is unnecessary to consider a scale of $\mathcal{H}$ (*i.e.*, a value of $\mathbb{R}_f$) at which the regions containing x and y are the same as those at the preceding scale. In other words, rather than considering the scales of $\mathbb{R}_f$ for which there is a global change in the hierarchy, one can focus on the scales for which the change of the hierarchy is local to x and y, *i.e.*, when the change involves a region containing either x or y.

Let f be a any map from E to $\mathbb{E}$ and let $\mathcal{H}$ be the quasi-flat zone hierarchy of f. Let x be any vertex of V and let us denote by $\mathcal{B}_{\mathcal{H}}(x)$ the set which contains every region R of the hierarchy $\mathcal{H}$ such that x belongs to R. The set $\mathcal{B}_{\mathcal{H}}(x)$ is called the *branch of* x *in*

$\mathcal{H}$. The *level* of a region R in $\mathcal{H}$, denoted by $level_{\mathcal{H}}(R)$, is the highest index of a partition that contains R in $\mathcal{H}$. The *(branch) range of* $\mathcal{H}$ *for* x, denoted by $\mathrm{brange}(f, x)$, is defined as the set that contains the level of every region of the branch of x in $\mathcal{H}$: $\mathrm{brange}(f, x) = \{level_{\mathcal{H}}(R) \mid R \in \mathcal{B}_{\mathcal{H}}(x)\}$. Let $u = \{x, y\}$ be any edge of T. We denote by $\mathbb{R}_f^u$ the series $(\lambda_0, \ldots, \lambda_\ell)$ of ordered values in $\mathrm{brange}(f, x) \cup \mathrm{brange}(f, y) \cup \{-\infty, \max\{\lambda \in \mathbb{E}\}\}$:

1. $\lambda_0 = -\infty$, and $\lambda_\ell = \max\{\lambda \in \mathbb{E}\}$;
2. $\{\lambda_i \mid i \in \{1, \ldots, \ell - 1\}\} = \mathrm{brange}(f, x) \cup \mathrm{brange}(f, y) \setminus \{\max\{\lambda \in \mathbb{E}\}\}$; and
3. for any i in $\{1, \ldots, \ell\}$, we have $\lambda_i > \lambda_{i-1}$.

Property 8. *Let f be any map from $E(T)$ in $\mathbb{E}$ and let $u = \{x, y\}$ be any edge in $E(T)$. Then, the series $\mathbb{R}_f^u$ is a stable bound series for (f, u).*

Proof. Since $\lambda_0 = -\infty$, since $\lambda_\ell = \max\{\lambda \in \mathbb{E}\}$, and since, for any $i \in \{1, \ldots \ell\}$ we have $\lambda_i < \lambda_{i-1}$, in order to establish Property 8, it is sufficient to prove that, for any i in $\{1, \ldots, \ell\}$, the interval $]\!]\lambda_{i-1}, \lambda_i]\!]_{\mathbb{E}}$ is a stable interval for (f, u). Let i be any element in $\{1, \ldots, \ell\}$. Let λ be any element in $]\!]\lambda_{i-1}, \lambda_i]\!]_{\mathbb{E}}$. By definition of $\mathbb{R}_f^u$, there exists j in $\{1, \ldots, \ell\}$ such that $\lambda_j = level(\mathcal{H}_x^\lambda)$ (resp. $\lambda_j = level(\mathcal{H}_y^\lambda)$). By definition of the level of a region, we deduce that $\lambda_j \geq \lambda$. Hence, we have $j \geq i$ which implies that $\lambda_j \geq \lambda_i$. Since $\mathcal{H}$ is a hierarchy, we have $\mathcal{H}_x^\lambda \subseteq \mathcal{H}_x^{\lambda_i} \subseteq \mathcal{H}_x^{\lambda_j}$ (resp. $\mathcal{H}_y^\lambda \subseteq \mathcal{H}_y^{\lambda_i} \subseteq \mathcal{H}_y^{\lambda_j}$). By definition of λ_j, we have $\mathcal{H}_x^\lambda = \mathcal{H}_x^{\lambda_j}$ (resp. $\mathcal{H}_y^\lambda = \mathcal{H}_y^{\lambda_j}$). Therefore, we also have $\mathcal{H}_x^\lambda = \mathcal{H}_x^{\lambda_i}$ (resp. $\mathcal{H}_y^\lambda = \mathcal{H}_y^{\lambda_i}$). Thus, $]\!]\lambda_{i-1}, \lambda_i]\!]_{\mathbb{E}}$ is a stable interval for (f, u). □

As a direct consequence of Property 8 and Theorem 5, we can state the following corollary which is the basis of a third algorithm for solving efficiently the minimization problem given in Equation (3.3). The difference with Corollary 7 is that the range of f (f being such that $\mathcal{H} = \mathcal{QFZ}(T, f)$) is replaced by the union of the branch ranges of $\mathcal{H}$ for x and for y.

Corollary 9. *Let f be any map from E in $\mathbb{E}$, let $\mathcal{H}$ be the quasi-flat zone hierarchy of G for f, let $u = \{x, y\}$ be any edge in E, and let $\mathbb{R}_f^u = (\lambda_0, \ldots, \lambda_\ell)$. Then, the following statement holds true:*

$$\lambda^\star_{\mathcal{H}}(\{x, y\}) = \min\{\max\{n_{\mathbb{E}}(\lambda_{i-1}), \hat{n}_{\mathbb{E}}\left(D(\mathcal{H}_x^{\lambda_i}, \mathcal{H}_y^{\lambda_i})\right)\} \mid i \in ind(\mathbb{R}_f^u), D(\mathcal{H}_x^{\lambda_i}, \mathcal{H}_y^{\lambda_i}) \leq \lambda_i\}.$$

Due to Corollary 9, to compute $\lambda^\star_{\mathcal{H}}(\{x, y\})$, it is sufficient to browse in increasing order the levels of the regions in the branches of x and of y until a value λ_i, such that $D(\mathcal{H}_x^{\lambda_i}, \mathcal{H}_y^{\lambda_i}) \leq \lambda_i$, is found. Finally, the value $\lambda^\star_{\mathcal{H}}(\{x, y\})$ is determined as the maximum of $\hat{\mathrm{n}}_{\mathbb{E}}\left(D(\mathcal{H}_x^{\lambda_i}, \mathcal{H}_y^{\lambda_i})\right)$ and $\mathrm{n}_{\mathbb{E}}(\lambda_{i-1})$, where $\{\lambda_0, \ldots \lambda_\ell\}$ is equal to $\mathbb{R}_f^{\{x,y\}}$. In order to propose such an algorithm, we need to browse in increasing order the levels of the regions

in the branches of x and of y. This can be done with a tree data structure, called a component tree, which represents the hierarchy. The component tree is used for various image processing tasks and is well studied in the field of mathematical morphology (see, *e.g.*, [Salembier et al., 1998] for their definition on vertex weighted graphs, [Cousty et al., 2013] for the case of edge-weighted graphs and quasi-flat zone, and [Perret et al., 2015] for their generalization to directed graphs). In classification, this tree is often called the dendrogram of the hierarchy.

As any tree, the component tree of $\mathcal{H}$ can be defined as a pair made of a set of nodes and of a binary (parent) relation on the set of nodes. More precisely, the *component tree of* $\mathcal{H}$ is the pair $\mathcal{T}_{\mathcal{H}} = (\mathcal{N}, parent)$ such that $\mathcal{N}$ is the set of all regions of $\mathcal{H}$ and such that a region R_1 in $\mathcal{N}$ is a *parent* of a region R_2 in $\mathcal{N}$ whenever R_1 is a minimal (for inclusion relation) proper superset of R_2. Note that every region in $\mathcal{N}$ has exactly one parent except the region V which has no parent and is called the *root* of the component tree of $\mathcal{H}$. Any region which is not the parent of another one is called a *leaf* of the tree. It can be observed that any singleton of V is a leaf of $\mathcal{T}_{\mathcal{H}}$ and that conversely any leaf of $\mathcal{T}_{\mathcal{H}}$ is a singleton of V.

In order to browse the branch of x in $\mathcal{H}$ from its component tree, it is enough to follow the next steps: (1) start with the node C that is the leaf $\{x\}$, (2) consider the parent of C, and (3) repeat step (2) until the root is found. Furthermore, it can be observed that the $level_{\mathcal{H}}$ attribute is increasing in the branch of x: for any non-root node C in $\mathcal{N}$, the level of the parent of C is never less than the level of C. Hence, the branch browsing process also allows browsing the branch range of $\mathcal{H}$ for x in increasing order. According to Corollary 9, in order to find the value $\lambda^{\star}_{\mathcal{H}}(\{x, y\})$, for any edge $\{x, y\}$ of T and any hierarchy $\mathcal{H}$, we have to consider the union of the ranges of $\mathcal{H}$ for x and for y, sorted in increasing order. This can be done by simultaneously browsing in the component tree $\mathcal{T}_{\mathcal{H}}$ the branches of x and of y. Algorithm 3 provides a precise description of a complete algorithm to find $\lambda^{\star}_{\mathcal{H}}(\{x, y\})$ using such a simultaneous branch browsing.

Algorithm 3: HGB Minimization by branch

Input : The component tree $\mathcal{T} = (\mathcal{N}, parent)$ of a hierarchy $\mathcal{H}$, an edge $u = \{x, y\}$ of T, an array $level$ that stores the level of every region of $\mathcal{H}$
Output: The value $\lambda^{\star}$ such that $\lambda^{\star} = \lambda^{\star}_{\mathcal{H}}(\{x, y\})$

$C_x := \{x\}$; $C_y := \{y\}$; $\lambda := \min(level[C_x], level[C_y])$; $\lambda_{prev} := -\infty$;
while $D(C_x, C_y) > \lambda$ **do**
  $\lambda_{prev} := \lambda$;
  $\lambda := \min(level[parent[C_x]], level[parent[C_y]])$;
  if $level[parent[C_x]] = \lambda$ **then** $C_x := parent[C_x]$;
  if $level[parent[C_y]] = \lambda$ **then** $C_y := parent[C_y]$;
end
$\lambda^{\star} := \max(\mathrm{n}_{\mathbb{E}}(\lambda_{prev}), \hat{\mathrm{n}}_{\mathbb{E}}\left(D(\mathcal{H}^{\lambda}_x, \mathcal{H}^{\lambda}_y)\right))$;

Individually, every instruction performed in Algorithm 3 has a constant time complexity. Therefore, in order to establish the overall time complexity of Algorithm 3, it is sufficient to bound the number of iterations of the main loop of Algorithm 3 (Line 2). It can be seen that the instructions and tests of this loop are executed at most $|\text{brange}(f, x)|+|\text{brange}(f, y)|$ times. In the worst case, at every level of the hierarchy the region containing x is merged with a singleton region. Hence, as there are $|V|$ vertices in G, in this case, the branch of x contains $|V|$ regions. Thus, the worst-case time complexity of Algorithm 3 is $O(|V|)$. It can be observed that the worst-case time complexity of Algorithm 3 is the same as the one of Algorithm 2. However, in many practical cases, the component tree of $\mathcal{H}$ is well balanced and each region of $\mathcal{H}$ results from the merging of two regions of (approximately) the same size. Then, if the tree is balanced, the branch of x contains $O(\log_2(|V|))$ nodes and the time-complexity of Algorithm 3 reduces to $O(\log_2(|V|))$ which is a significant improvement compared to Algorithm 2. Such improvement is verified in terms of execution times in Section 4.4.

4.3 Hierarchy update for HGB method

In this section, we focus on Lines 3 and 6 of Method 1, that is, on computing the quasi-flat zone hierarchy of a weight map f. This computation is repeated at every iteration of the method (*i.e.*, for every edge of the tree T). Hence, finding an efficient way to perform this task in the context of Method 1 presents a high speedup potential.

4.3.1 Naive update

The straightforward implementation of Lines 3 and 6 of Method 1 consists of computing, at each iteration, the quasi-flat zone hierarchy of f using an efficient algorithm such as the one presented in [Najman et al., 2013]. Provided that the edges of T can be sorted in linear time, which, in our case, can be done with a counting sort algorithm, the time-complexity of this algorithm is quasi-linear $O(|V|+\alpha(|V|))$ (where α is the extremely slowly growing inverse of the single-valued Ackermann function). Hence, in this case, the overall time-complexity of Lines 3 and 6 is $O(|V|^2 + |V|\alpha(|V|))$ since the quasi-flat zone hierarchy computation is repeated exactly $|E(T)| = |V| - 1$ times.

However, from one iteration of the main loop of Method 1 to the next one, only

the weight of one edge of the graph is updated and therefore most parts of the component tree remain unchanged. Therefore, an important speedup can be obtained if we avoid to recompute from scratch the whole component tree at each iteration. In order to avoid this recomputation, we need to rely on an algorithm that only updates the part of the component tree which is affected by the single weight update considered at the present iteration. In this section, we propose such an algorithm that is referred to as an incremental quasi-flat zone update algorithm.

4.3.2 Incremental update with component tree

The presented incremental quasi-flat zone update algorithm relies on works done for parallel computation of component trees [Wilkinson et al., 2008, Havel et al., 2019, Götz et al., 2018, Carlinet and Géraud, 2014]. In these articles, the authors presented algorithms to merge the component trees of two disjoint (adjacent) image blocks in order to obtain the component tree of the image consisting of these two blocks. We can adapt these algorithms (in particular, Algorithm 6 in [Havel et al., 2019]) into an incremental quasi-flat zone update algorithm. At each iteration of Line 3 in Method 1, the weight of the edge u is decreased from its initial value $\max\{\lambda \in \mathbb{E}\}$ to its final value resulting from the minimization of Equation (3.3). This means that, before we decrease the weight of $u = \{x, y\}$, the components containing x and y in the tree were disjoint (up to level $\max\{\lambda \in \mathbb{E}\}$). We can adapt the algorithm proposed by Havel et al. [2019], in order to merge these disjoint parts of the tree and update the tree only on the components containing x and y, thus avoiding the need to recompute the whole hierarchy at every iteration of Method 1.

Algorithm 4 gives a precise description of this quasi-flat zone update algorithm given the component tree $\mathcal{T}$ of a hierarchy $\mathcal{H}$ which is the quasi-flat zone hierarchy of a weight map f, the edge $u = \{x, y\}$ whose weight must be decreased, and the value λ which corresponds to the decreased weight of u. The algorithm first identifies the part of the tree which must be modified, namely the components containing x and y at levels higher than λ (Line 1). Then, the tree representation of the components containing x and y at higher level is built (Lines 2 to 14) by either merging existing nodes (Line 10), creating new parenthood relation between existing nodes (Lines 7 and 11), or creating new nodes (Line 2). To perform these tasks, Algorithm 4 relies on four auxiliary functions.

- Given a vertex x of V and a level λ in $\mathbb{E}$, $findTransition(x, \lambda)$ returns a node n which is the ancestor of the node representing $\{x\}$ such that (i) $level[n] \leq \lambda$, and (ii) $level[parent[n]] > \lambda$. This operation is performed by traversing upward the

branch of $\mathcal{T}$ containing x, starting at the node $n = \{x\}$ and ending when a node n satisfying (i) and (ii) is found.

- Given two nodes c_1 and c_2 of $\mathcal{T}$ and a value λ in $\mathbb{E}$, $node(c_1, c_2, \lambda)$ creates a node n at level λ which becomes the parent of c_1 and of c_2.
- Given two nodes c_1 and c_2 of $\mathcal{T}$, $attach(c_1, c_2)$ sets c_1 to be the parent of c_2.
- Given two nodes c_1 and c_2, $merge(c_1, c_2)$ calls $merge(c_2, c_1)$ if c_2 has more children than c_1, otherwise it sets the parent of the children of c_2 to be c_1, and it returns c_1.

Algorithm 4: Incremental quasi-flat zone hierarchy update

Input : The component tree $\mathcal{T} = (\mathcal{N}, parent)$ of the hierarchy $\mathcal{H}$ which is the quasi-flat zone hierarchy of a map f, an array $level$ that stores the level of every node of $\mathcal{T}$ (*i.e.*, every region of $\mathcal{H}$), an edge $u = \{x, y\}$ of T, the value λ at which the weight of the edge u must be decreased.

Output: The updated component tree $\mathcal{T}$, which is the component tree of the quasi-flat zone hierarchy of the updated map f (*i.e.*, the map f' such that $f'(v) = f(v)$ for any $v \neq u$ and $f'(u) = \lambda$)

```
1  c1 := findTransition(x, λ) ; c2 := findTransition(y, λ) ;
2  n := node(c1, c2, n_E(λ)) ;
3  c1 := parent[c1]; c2 := parent[c2];
4  do
5      if level[c2] < level[c1] then swap(c1, c2);
6      if level[c1] < level[c2] then
7          attach(c1, n) ;
8          n := c1; c1 := parent[c1] ;
9      else
10         n' := merge(c1, c2) ;
11         attach(n', n) ;
12         n := n'; c1 := parent[c1]; c2 := parent[c2];
13     end
14 while p1 ≠ p2 ;
```

Algorithm 4 modifies the tree structure in the following manner: first, given an edge $u = \{x, y\}$ of decreased weight λ, it starts from the singleton components $\{x\}$ and $\{y\}$. Then, $findTransition$ identifies the nodes c_1 and c_2 associated to the components $\mathcal{H}_x^\lambda$ and $\mathcal{H}_y^\lambda$, respectively. A node n is created to represent the union of theses components (Line 2). Then, the do-while loop (Lines 5 to 14) traverses simultaneously the branches containing x and y from the nodes c_1 and c_2, identifying the ancestors of these nodes, and updating the parenthood relationships along these branches. At each iteration, the two nodes are merged if they have the same level (Line 10) or, if one has a level less than the other, the one of highest level becomes the parent of the one of lowest level (Line 7). This is repeated until a common ancestor is found. Consequently, only the components

containing x and y are involved in the update algorithm and we do not need to recompute a whole hierarchy at every iteration.

It can be seen that Algorithm 4 involves only constant time operations performed on the nodes corresponding to the branch containing x and y. Therefore, following the discussion at the end of Section 4.2.4, in the worst case, Algorithm 4 runs in $O(|V|)$ time complexity and in the case where the tree $\mathcal{T}$ is balanced it runs in $O(\log_2(|V|))$ time complexity. Hence, in the worst case, using Algorithm 4, the overall time-complexity of Line 3 in Method 1 is $O(|V|^2)$, whereas in the most favorable case where the tree remains balanced, this complexity reduces to $O(|V|\log_2|V|)$, for computing the result of the minimization problem in the HGB method.

4.4 Experiments

The experiments reported in this section aim at measuring and comparing the execution times of all the variations of the algorithms proposed in the previous sections for the HGB method. The experimental set-up is first presented in Section 4.4.1 and then the experimental results are given in Section 4.4.2.

4.4.1 Experimental set-up

As we have three variations for the minimization step (Line 4 in Method 1), namely Algorithms 1, 2 and 3, and two variations for the quasi-flat zone hierarchy computation (Lines 3 and 6 in Method 1), namely the non-incremental one [Najman et al., 2013] and the incremental one (Algorithm 4), the total number of all the combinations is six. Hence, in total, we studied the execution times of these six variations.

All the algorithms were implemented in C and executed on a computer with a 3.2 GHz CPU, 8GB RAM on Ubuntu Linux 16.04.

In order to cope with realistic situations, we used the Berkeley Segmentation Dataset (BSDS) [Arbelaez et al., 2011] for our experiments. This dataset consists of 500 natural images of size 321×481 pixels and is very popular in image segmentation experiments.

A first assessment consists of measuring the execution times of the six presented variations when applied to one of the images from the BSDS dataset. The chosen image is

shown in Figure 4.2(a) and the result of any of the six variations is shown in Figure 4.2(b) in the form of a saliency map. This image was chosen because of its textured aspect which leads to hierarchies with a high number of regions and scales, hence exploiting the ability of the algorithms to deal with a high number of regions and of levels (5218 levels). This experiment is designed in order to assess the gain achieved from switching from the least efficient variation of Method 1 to the most efficient one.

(a) (b)

Figure 4.2: (a) Original image, from the BSDS dataset, used for the algorithm assessment, and, (b) the resulting HGB hierarchy represented as a saliency map.

A second assessment is designed in order to assess the scalability of the most efficient variation of Method 1 when applied to a whole dataset of images representing a wider variety of situations that can be encountered in computer vision tasks. This second experiment consists of measuring the execution times taken by the fastest variation of Method 1 on the full BSDS dataset. The fastest variation to compute the result of Method 1 is determined from the first assessment and consists of using Algorithm 3 and Algorithm 4.

Reported execution times result from repeating ten times the execution of a same method on a same image and considering the average time taken by these ten executions.

Table 4.1: Execution times from the image of Figure 4.2(a) (321 × 481 pixels). The resulting hierarchy contains 5218 levels.

Hierarchy Update	Minimization Algorithm	Execution times (seconds)		
		Total	QFZ	Minimization
Non-Incremental	Algorithm 1	14666.08	13186.31	1479.56
	Algorithm 2	13392.51	13375.29	17.02
	Algorithm 3	13166.25	13165.54	0.49
Incremental	Algorithm 1	1487.96	0.13	1487.75
	Algorithm 2	15.42	0.13	15.21
	Algorithm 3	0.49	0.10	0.32

4.4.2 Experimental results

Table 4.1 shows the execution times taken by each of the six variations of Method 1 when applied to the image shown in Figure 4.2(a). In the table, the first column labeled *QFZ algorithm* refers to the type of algorithm used to construct the QFZ hierarchy: it can be either the non-incremental one based on [Najman et al., 2013] or the incremental one, namely Algorithm 4, based on [Havel et al., 2019] and on [Wilkinson et al., 2008]. The second column refers to the algorithm which is used for performing the minimization described in Equation (3.3): it can be either the naive algorithm (Algorithm 1), the minimization by range algorithm (Algorithm 2) or the minimization by branch algorithm (Algorithm 3). The third column presents the overall execution times of the six variations which include the time for computing quasi-flat zone hierarchies (fourth column) and for computing the result of the minimization described by Equation (3.3) (fifth column).

As we can observe in Table 4.1, the total execution time, using the non-incremental QFZ hierarchy construction and any minimization algorithm, leads into very prohibitive times of over four hours. However, most of this time is consumed on the QFZ hierarchy construction (over three hours). For the minimization step, Algorithm 1 was the least efficient taking around 24 minutes. Then, Algorithm 2 using the range minimization leads to 17 seconds of execution. Finally, Algorithm 3 was the fastest algorithm for the minimization step with less than one second. When we used the incremental QFZ hierarchy construction, the time spent on updating the hierarchical tree took only 0.1 seconds. This, with Algorithm 3 (minimization step), lead to a total execution time of 0.49 seconds, which is our most efficient variation. From these results, we can conclude that it is only possible to compute the HGB method in user time with the help of both the incremental QFZ hierarchy construction and the minimization by branch (Algorithm 3).

Figure 4.3 shows the distribution of the execution times of our most efficient variation of Method 1 applied to all images in the BSDS dataset. The average execution time over the dataset is 0.47 seconds with a standard deviation of 0.09 seconds. This confirms

that our most efficient variation of Method 1, namely the implementation of Method 1 with Algorithm 3 and Algorithm 4, runs in user time whatever the considered image from the BSDS dataset.

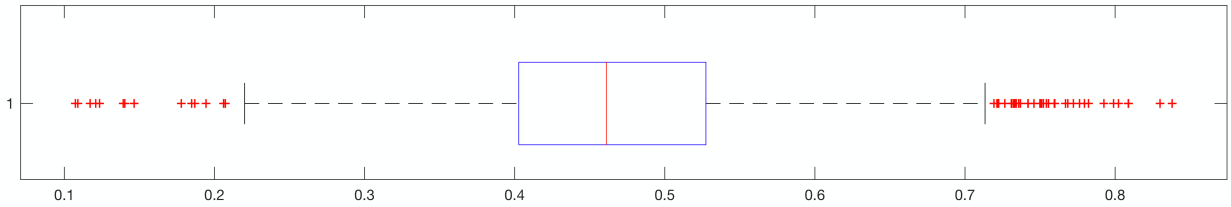


Figure 4.3: Box and Whisker plot for execution times on BSDS dataset. On the horizontal axis the times are given in seconds.

4.5 Concluding remarks

In this chapter, we investigated the algorithmic challenges involved in implementing the HGB method proposed in [Guimarães et al., 2017]. With the aim of developing exact and time-efficient algorithms for its implementation on images. We focused on the two main steps of the HGB method for improving efficiency: (i) the minimization involved in Equation (3.3), and (ii) the computation of the quasi-flat zones hierarchies. Concerning (i), we presented a general framework which allows reducing the search space involved in the minimization problem as shown by Theorem 5. We considered two applications of this framework leading to two algorithms which both improve the efficiency, in terms theoretical of time complexity and of practical running times, compared to a naive algorithm for solving the minimization problem. Furthermore, due to the proposed framework, the proof of correctness for each of these two algorithms (namely Corollaries 7 and 9) was provided.

In order to compute efficiently the quasi-flat zone hierarchy (ii), we considered a non-incremental and an incremental algorithm based on [Najman et al., 2013] and on [Havel et al., 2019], respectively. Even if the worst-case complexities of these two algorithms are comparable, the running times of the HGB method are very significantly decreased when the incremental algorithm is used instead of the non-incremental one. Overall, on images from the standard BSDS dataset, the least efficient strategy that we proposed computes the result of the HGB method in more than four hours whereas the most efficient one takes about half a second.

Furthermore, we would like to emphasize that the framework that we introduced in this chapter lead to a better understanding of the minimization equation which is at

the heart of the HGB method. The contributions we made on this chapter allow us to fulfill objectives 1, 2 and 3 (see Chapter 1). About objective 2, we elaborated a precise formalism to study properties to hierarchize the GB method. Based on this theoretical background, we accomplished objective 1 by proposing a HGB method, where one level of the resulting hierarchy corresponds to one instance of the flat GB segmentation problem. We have also established the theoretical background to develop efficient algorithms to compute the HGB method, thus fulfilling objective 3.

We must acknowledge that our new understanding of the GB and HGB method opened doors towards modifications of the minimization equation which could allow us to compute hierarchies with characteristics different to the ones originally proposed by Guimarães et al. [2017]. Indeed, a better understanding of the minimization equation allowed us to determine that other scales different to the one obtained originally by the minimization equation that can lead to different segmentation results. Furthermore, the result of the studies from this chapter also allowed us to have an algorithm that efficiently computes all the space of observation scales. In the next chapter we study new strategies to select different observation scales from this space.

Chapter 5

Hierarchical segmentation from a non-increasing edge observation attribute

5.1 Introduction

In this chapter, we study the selection strategy originally proposed for the HGB method and provide a novel analysis on this selection leading to (1) a better understanding of the method and to (2) several new HGB variants which are shown to provide better scores in our assessments measures. Let us recall that in Chapter 3, we presented the original HGB method, and we introduced some notions such as the level of a segmentation in a hierarchy, which is also called an *observation scale*. Guimarães et al. [2017] proposed a hierarchical graph-based image segmentation (HGB) method inspired by the Felzenszwalb-Huttenlocher dissimilarity measure [Felzenszwalb and Huttenlocher, 2004] (see Chapter 3). Originally, the HGB method computes for each edge of a graph, the minimum observation scale in a hierarchy at which two regions linked by this edge should merge according to the dissimilarity. In this chapter, we make a new analysis of this strategy.

We begin this new analysis in Section 5.2, where we present an explicit definition of the edge observation attribute and a Boolean criterion which are at the basis of the HGB method and which makes it possible to select an observation scale for each edge of the graph. We show that this attribute and the related criteria are not increasing with respect to scales. Furthermore, we notice that the HGB method handles this non-increasing behavior with a simple rule, namely the min-decision rule, known in mathematical morphology since the work of Salembier et al. [1998]. Some more elaborated strategies have been proposed to deal with attributes that are not increasing with respect to scales [Xu et al., 2016, Salembier and Liesegang, 2018]. These strategies are very successful in applications of image analysis. Following these trends, we proposed several new strategies to handle the non-increasing behavior of the observation attribute and criterion behind the HGB method.

In this sense, we present in Section 5.3, how to compute all the intervals of scales

at which the boolean criterion holds true. We call these observation intervals, we give a formal definition of them and we also introduce an algorithm to compute them. In Section 5.4, we introduce new strategies to select observation scales. These strategies allow us to select an observation scale for each edge of the graph. Intuitively, they enable us to filter the set of all the scales for which the Boolean observation criterion holds true, before selecting one scale, with a simple strategy (such as the min-decision rule), among the scales remaining after the filtering. Such filtering is also done with rank and connected filters [Salembier and Wilkinson, 2009]. We present our new selection and filtering strategies in Subsections 5.4.1 and 5.4.2.

Using the segmentation evaluation framework proposed by Perret et al. [2018], we show in Section 5.5, that the newly observation scale selection strategies significantly outperform the original HGB method on Pascal VOC 2010 and 2012 datasets [Everingham et al., 2010, 2012]. Finally, in Section 5.6, we present some concluding remarks from the contributions we made on this chapter and how they relate to the objectives of this thesis. We must point out that one of the contributions we made on this chapter, involves the identification of key hyper-parameters of the HGB method (observation scales) and new strategies to set up these hyper-parameters. This chapter consists of the work presented in [Cayllahua-Cahuina et al., 2020, 2019a].

5.2 Non-increasing observation attribute and criterion

In this section, we provide a formal definition of the observation attribute and criterion involved in Equation (3.3). Then, we discuss their non-increasing behavior opening the doors towards new strategies to select observation scale values based on Felzenszwalb-Huttenlocher dissimilarity measure as used in Method 1.

Let us recall some of the notions we presented in Chapter 2. We consider an edge-weighted graph (G, w) and whose corresponding minimum spanning tree T of G is taken by the HGB method as input to compute a new weight map f; which leads to a new hierarchy $\mathcal{H} = \mathcal{QFZ}(T, f)$, where $\mathcal{QFZ}(T, f)$ is the quasi-flat zone hierarchy of T for f (see Subsection 2.2.5 in Chapter 2). In the remaining part of this section, we consider that $\mathcal{H}$ is such hierarchy and that $u = \{x, y\}$ is any edge of T. The weight map f is computed as a result of performing the minimization presented in Equation (3.3) for each edge u, the search space of this minimization takes place in the finite set of considered level values, that is the subset $\mathbb{E}$ of $\mathbb{R}$.

The *edge observation attribute (of* $u = \{x, y\}$ *in* $\mathcal{H}$*)* is the map $\mathcal{A}$ from $\mathbb{E}$ into $\mathbb{R}$

defined by:

$$\mathcal{A}(\lambda) = \lambda - D(\mathcal{H}_x^\lambda, \mathcal{H}_y^\lambda), \tag{5.1}$$

for any λ in $\mathbb{E}$, where D is the dissimilarity measure between regions (see Subsection 3.3 in Chapter 3). The *edge observation criterion (of u in $\mathcal{H}$)* is the map $\mathcal{C}$ from $\mathbb{E}$ in the set $\{true, false\}$ such that $\mathcal{C}(\lambda) = true$ if $\mathcal{A}(\lambda)$ is greater than or equal to 0; otherwise $\mathcal{C}(\lambda) = false$. Let λ be any element in $\mathbb{E}$, we say that λ is a *positive observation scale (of u in $\mathcal{H}$)* whenever $\mathcal{C}(\lambda)$ holds true. On the contrary, if λ is not a positive observation scale, then we say that λ is a *negative observation scale (of u in $\mathcal{H}$)*.

Observe that the value $\lambda^\star_\mathcal{H}(x, y)$ defined in Equation (3.3) is simply the lowest element of $\mathbb{E}$ such that $\mathcal{C}(\lambda)$ is true. In other words, $\lambda^\star_\mathcal{H}(x, y)$ is the lowest positive observation scale of $u = \{x, y\}$ in $\mathcal{H}$. Additionally, the largest negative observation scale is defined by:

$$\overline{\lambda}^\star_\mathcal{H}(x, y) = \max\left\{\lambda \in \mathbb{E} \mid D\left(\mathcal{H}_x^\lambda, \mathcal{H}_y^\lambda\right) > \lambda\right\}. \tag{5.2}$$

Intuitively, a positive observation scale corresponds to a level of the hierarchy $\mathcal{H}$ for which the two regions $\mathcal{H}_x^\lambda$ and $\mathcal{H}_y^\lambda$ linked by $u = \{x, y\}$ should be merged according to the observation criterion $\mathcal{C}$. On the other hand, a negative observation scale corresponds to a level of the hierarchy for which the two associated regions should remain disjoint. Furthermore, for a positive observation scale λ, the observation attribute $\mathcal{A}(\lambda)$ can be intuitively interpreted as a measure of the strength at which the two associated regions should be merged whereas, for a negative observation scale, the observation attribute can be seen as the opposite of the strength at which the two regions should remain disjoint: when the observation attribute is highly positive, the two regions should be merged with a high confidence whereas when the observation attribute is negative and low, the two regions should remain disjoint with a high confidence.

A desirable property would be for the observation criterion $\mathcal{C}$ to be increasing with respect to scales. A Boolean criterion $\mathcal{C}$ is considered increasing if for any scale value $\lambda \in \mathbb{E}$, if $\mathcal{C}(\lambda)$ holds true, then $\mathcal{C}(\lambda')$ is true for any scale λ' greater than λ. Indeed, in such desirable case, any level in $\mathbb{E}$ greater than $\lambda^\star_\mathcal{H}(x, y)$ would be a positive observation scale, whereas any level not greater than $\lambda^\star_\mathcal{H}(x, y)$ would be a negative scale. In other words, we would have $\lambda^\star_\mathcal{H}(x, y) = \mathrm{n}_\mathbb{E}\left(\overline{\lambda}^\star_\mathcal{H}(x, y)\right)$. Hence, it would be easily argued that the observation scale of the edge u must be set to $\lambda^\star_\mathcal{H}(x, y)$. However, in general, the observation attribute $\mathcal{A}$ and thus the observation criterion $\mathcal{C}$ are not increasing (see a counterexample in Figure 5.1) and we have $\lambda^\star_\mathcal{H}(x, y) < \mathrm{n}_\mathbb{E}\left(\overline{\lambda}^\star_\mathcal{H}(x, y)\right)$. Therefore, it is interesting to investigate strategies which can be adopted to select a significant observation scale between $\lambda^\star_\mathcal{H}(x, y)$ and $\overline{\lambda}^\star_\mathcal{H}(x, y)$. A graphical illustration of different situations which may occur for a given edge u and hierarchy $\mathcal{H}$ is presented in Figure 5.1. In other words, it is interesting to investigate strategies to transform the observation attribute $\mathcal{A}$ and criterion $\mathcal{C}$ into an increasing attribute $\mathcal{A}'$ and an increasing criterion $\mathcal{C}'$.

In the framework of mathematical morphology, non-increasing regional attributes/criteria are known to be useful but difficult to handle. Several rules or strategies to handle non-increasing criteria have been considered in the context of connected filters. Among them, one may cite the min- and max-rules [Salembier et al., 1998], the Vitterbi [Salembier et al., 1998] and shape-space filtering strategies [Xu et al., 2016], and the max-tree pruning strategies based on graph signal processing [Salembier and Liesegang, 2018]. Note that the strategy adopted in Equation (3.3) corresponds to the min-rule and that the strategy consisting of selecting $\overline{\lambda}^{\star}_{\mathcal{H}}(x, y)$ corresponds to the max-rule. In this part of our research, our main goal is to investigate other strategies to efficiently handle the non-increasing observation criterion $\mathcal{C}$ in the context of hierarchical segmentation based on the Felzenszwalb-Huttenlocher region dissimilarity measure (see Equation 3.2). These strategies are based on the analysis of all positive and negative observation scales and their associated attribute values. Therefore, before presenting these strategies, it is necessary to provide a method to obtain all positive and negative observation scales.

5.3 Observation intervals: definition and algorithm

As established in the previous section, the observation attribute and observation criterion (obtained by thresholding the observation attribute at value 0) at the basis of the HGB method are not increasing. Hence, in order to find the set of all scales for which the criterion holds true (*i.e.*, the scales for which the regions linked by an edge u should

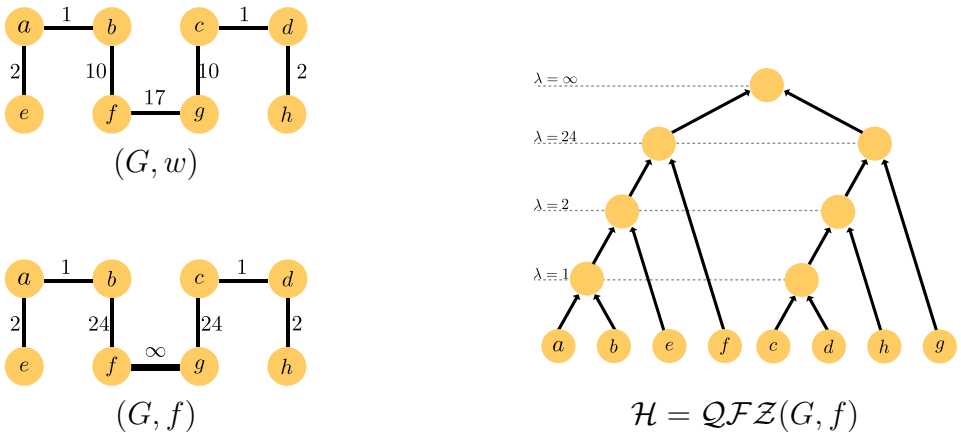


Figure 5.1: Counterexample for the increasing property of the observation attribute $\mathcal{A}$ and criterion $\mathcal{C}$: for the edge $u = \{x, y\}$ with $x = f$, $y = g$, we have $D(\mathcal{H}^1_x, \mathcal{H}^1_y) = 17$, $D(\mathcal{H}^{23}_x, \mathcal{H}^{23}_y) = 17$, and $D(\mathcal{H}^{25}_x, \mathcal{H}^{25}_y) = 28$. Hence, we have $\mathcal{A}(1) = 1 - 17 = -16$, $\mathcal{A}(23) = 23 - 17 = 6$, and $\mathcal{A}(25) = 25 - 28 = -3$, so that $\mathcal{C}(1) = false$, $\mathcal{C}(23) = true$, and $\mathcal{C}(25) = false$, which shows that $\mathcal{A}$ and $\mathcal{C}$ are not increasing.

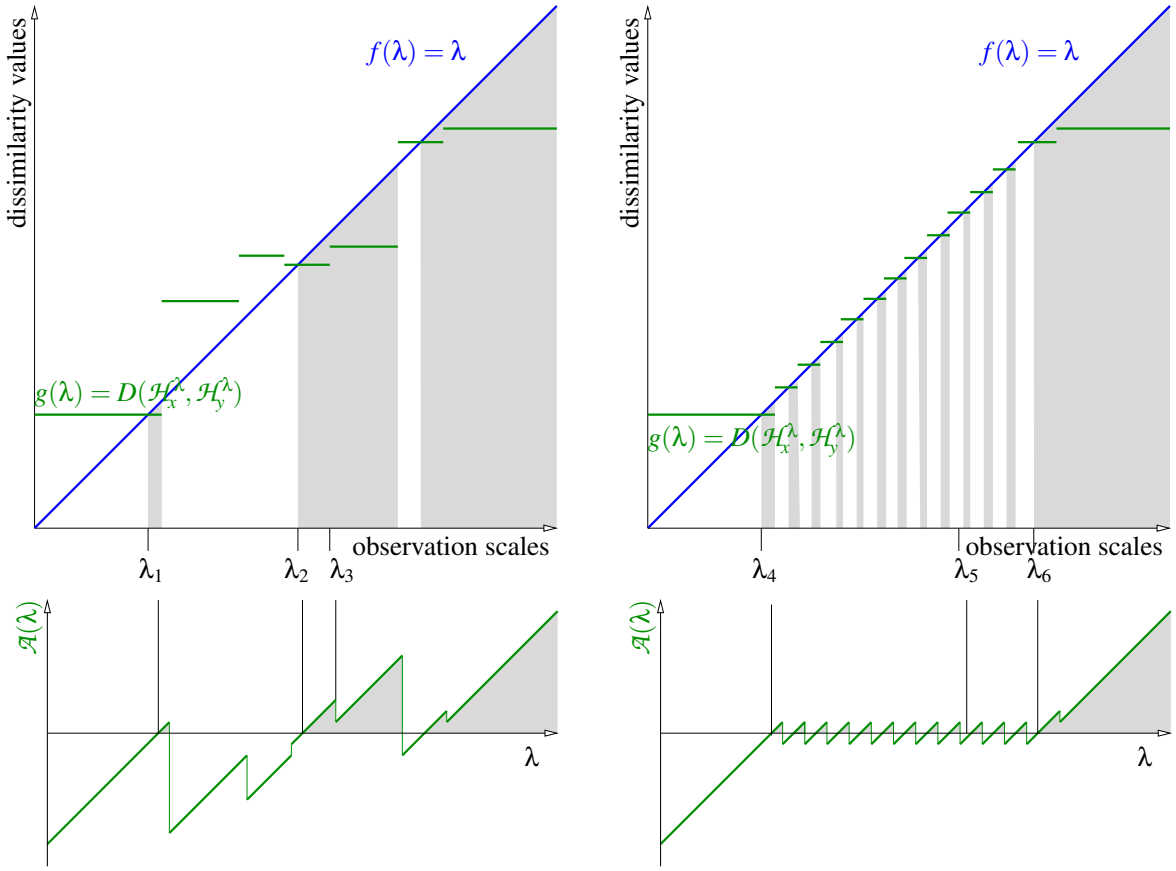


Figure 5.2: Illustration of the observation attribute and criterion and of possible observation scale selection strategies. The positive observation intervals (the intervals in which the observation criterion holds true) are represented in gray. On the top, the dissimilarity measure $D(\mathcal{H}_x^\lambda, \mathcal{H}_y^\lambda)$ is depicted in green as a function of λ, whereas, on the bottom, the observation attribute $\mathcal{A}(\lambda)$ is depicted in green as a functions of λ. On the left, the min-, the lower-length and the lower p-rank selection strategies select the scales λ_1, λ_2 and λ_3, respectively (for a length threshold which is a little larger than the leftmost gray interval and for $p = 0.3$), whereas, on the right, the max-, the upper-length and the upper p-rank selection strategies select the scales λ_6, λ_4 and λ_5, respectively. Note that, for the sake of readability, the values of the functions are depicted for all scales taken in $\mathbb{R}$, whereas the definitions are given only for a finite subset $\mathbb{E}$ of $\mathbb{R}$.

be merged), it is not enough to find the minimum of the positive observation scales. In Chapter 4, an algorithm to find the minimum observation scale such that the observation criterion holds true is given. Instead, in this section, an algorithm to find all scales at which the observation criterion holds true is provided. The proposed algorithm searches for the bounds of all (maximal) scale intervals for which the criterion holds true. Before presenting the algorithm that computes the bounds of all these intervals, we start with the definition of such intervals, called positive observation intervals.

Let λ_1 and λ_2 be any two real numbers in $\mathbb{E} \cup \{-\infty\}$ such that $\lambda_1 < \lambda_2$. We remind the reader that in Equation 4.1, $]\!]\lambda_1, \lambda_2]\!]_{\mathbb{E}}$ denotes the subset of $\mathbb{E}$ that contains every element of $\mathbb{E}$ that is both greater than λ_1 and not greater than λ_2: $]\!]\lambda_1, \lambda_2]\!]_{\mathbb{E}} = \{\lambda \in$

$\mathbb{E} \mid \lambda_1 < \lambda \leq \lambda_2\}$. Furthermore, any subset I of $\mathbb{E}$ is called an open-closed interval of $\mathbb{E}$, or simply an interval, if there exist two real values λ_1 and λ_2 in $\mathbb{E}$ such that I is equal to $]\!]\lambda_1, \lambda_2]\!]_{\mathbb{E}}$.

Definition 10 (observation interval)**.** *Let $\mathcal{H}$ be any hierarchy, let u be any edge in the edge weight map of T ($E(T)$), and let I be an interval. We say that I is a* positive observation interval (resp. a negative observation interval) for $(\mathcal{H}, u)$ *if the two following statements hold true:*

1. *any element in I is a positive (resp. negative) observation scale for $(\mathcal{H}, u)$; and*
2. *I is maximal among all intervals for which statement (1) holds true,* i.e.*, any interval which is a proper superset of I contains a negative (resp. positive) observation scale for $(\mathcal{H}, u)$.*

The set of all positive (resp. negative) observation intervals is denoted by $\Lambda_{\mathcal{H}}(u)$ (resp. by $\overline{\Lambda}_{\mathcal{H}}(u)$) .

In order to compute $\Lambda_{\mathcal{H}}(\{x, y\})$, we follow the strategy presented in Chapter 4, which relies on the *component tree* of the hierarchy $\mathcal{H}$. The *component tree of* $\mathcal{H}$ is the pair $\mathcal{T}_{\mathcal{H}} = (\mathcal{N}, parent)$ such that $\mathcal{N}$ is the set of all regions of $\mathcal{H}$ and such that a region R_1 in $\mathcal{N}$ is a *parent* of a region R_2 in $\mathcal{N}$ whenever R_1 is a minimal (for inclusion relation) proper superset of R_2. Note that every region in $\mathcal{N}$ has exactly one parent, except the region V which has no parent and is called the *root* of the component tree of $\mathcal{H}$. Any region which is not the parent of another one is called a *leaf* of the tree. It can be observed that any singleton of V is a leaf of $\mathcal{T}_{\mathcal{H}}$ and that conversely any leaf of $\mathcal{T}_{\mathcal{H}}$ is a singleton of V. The *level* of a region R in $\mathcal{H}$ is the highest index of a partition that contains R in $\mathcal{H}$. Then, the proposed algorithm, whose precise description is given in Algorithm 5, browses in increasing order the levels of the regions containing x and y until finding a value λ such that $D(\mathcal{H}^{\lambda}_x, \mathcal{H}^{\lambda}_y) \leq \lambda$, or, in other words, such that $\mathcal{A}(\lambda) > 0$ and hence, $\mathcal{C}(\lambda) = true$. This value is then $\lambda^{\star}_{\mathcal{H}}(x, y)$ defined by Equation (3.3). This value is also the lower bound of the first positive observation interval. If we keep browsing the levels of the regions containing x and y in this tree, as long as $D(\mathcal{H}^{\lambda}_x, \mathcal{H}^{\lambda}_y) \leq \lambda$, we can identify the upper bound of this first positive observation interval. We can further continue to browse the levels of the regions containing x and y in the tree in order to identify all positive observation intervals. Therefore, at the end of the execution, we can return the set $\Lambda_{\mathcal{H}}(\{x, y\})$ of all positive observation intervals. From the set $\Lambda_{\mathcal{H}}(\{x, y\})$, we can obtain by duality the set $\overline{\Lambda}_{\mathcal{H}}(\{x, y\})$ of all negative observation intervals.

The time complexity of Algorithm 5 depends linearly on the number of regions in the branches of the component tree of $\mathcal{H}$ containing x and y since it consists of browsing all these regions from the leaves to the root. In the worst case, at every level of the hierarchy the region containing x is merged with a singleton region. Hence, as there are $|V|$ vertices

Algorithm 5: Positive observation intervals

Input : The component tree $\mathcal{T} = (\mathcal{N}, parent)$ of a hierarchy $\mathcal{H}$, an edge $u = \{x, y\}$ of the minimum spanning tree associated to T, an array $level$ that stores the level of every region of $\mathcal{H}$

Output: A set L containing all elements of $\Lambda_{\mathcal{H}}(\{x, y\})$

$C_x := \{x\}$; $C_y := \{y\}$; $\Lambda_{\mathcal{H}}(\{x, y\}) = \{\}$;
$\lambda := \min(level[C_x], level[C_y])$; $\lambda_{prev} := -\infty$;
do
  // Search the lower bound of the next positive interval. **while** $D(C_x, C_y) > \lambda$ **do**
    $\lambda_{prev} := \lambda$;
    $\lambda := \min(level[parent[C_x]], level[parent[C_y]])$;
    if $level[parent[C_x]] = \lambda$ **then** $C_x := parent[C_x]$;
    if $level[parent[C_y]] = \lambda$ **then** $C_y := parent[C_y]$;
  end
  $\lambda_{lower} := \max(\mathrm{n}_{\mathbb{E}}(\lambda_{prev}), \hat{\mathrm{n}}_{\mathbb{E}}\left(D(\mathcal{H}^{\lambda}_{x}, \mathcal{H}^{\lambda}_{y})\right))$;
  // Search the upper bound of the next positive interval. **while** $D(C_x, C_y) \leq \lambda$ ***and*** $parent[C_x] \neq root$ ***and*** $parent[C_y] \neq root$ **do**
    $\lambda_{prev} := \lambda$;
    $\lambda := \min(level[parent[C_x]], level[parent[C_y]])$;
    if $level[parent[C_x]] = \lambda$ **then** $C_x := parent[C_x]$;
    if $level[parent[C_y]] = \lambda$ **then** $C_y := parent[C_y]$;
  end
  $\lambda_{upper} := \max(\mathrm{n}_{\mathbb{E}}(\lambda_{prev}), \hat{\mathrm{n}}_{\mathbb{E}}(\lambda_{prev}))$;
  $L.add(\llbracket \lambda_{lower}, \lambda_{upper} \rrbracket)$;
while $parent[C_x] \neq root$ ***and*** $parent[C_y] \neq root$;

in G, in this case, the branch of x contains $|V|$ regions. Thus, the worst-case time complexity of Algorithm 5 is $O(|V|)$. However, in many practical cases, the component tree of $\mathcal{H}$ tends to be balanced and each region of $\mathcal{H}$ results from the merging of two regions of (approximately) the same size. Then, if the tree is balanced, the branch of x contains $O(\log_2(|V|))$ nodes and the time-complexity of Algorithm 5 reduces to $O(\log_2(|V|))$.

5.4 Selecting observation scales

In this section, we investigate selection strategies to choose an observation scale from the sets of positive (respectively negative) observation intervals, *i.e.*, the set of scales at which the observation criterion holds true (respectively false). In Section 5.4.1, we present the min-, max-, and rank-rules to select observation scales from either the set of

positive or negative observation scales. Note that the min- and max-rules correspond to simple strategies known in mathematical morphology to prune the branch of a tree (aka a hierarchy) based on non-increasing regional attributes and criteria whereas the rank rules extend them to take a more robust decision. Then, in Section 5.4.2, we propose to improve the results obtained with these rules by pre-filtering out positive or negative observation intervals which might be considered as irrelevant with respect to a given attribute, before applying the min-, max- or rank-rules. Whereas, the selection rules can be seen as a way to transform a non-increasing criterion into an increasing one, the filtering strategies can be seen as regularization strategy for the non-increasing attribute on which the (observation) criterion is based. Overall, the proposed strategy allows us to transform a non-increasing attribute-based criterion into an increasing one based on a regularized version of the observation attribute.

5.4.1 Min-, max- and rank-selection rules

In this section, the symbol K stands for a finite set of observation scales (either negative or positive) and we define several useful rules to pick one observation scale in such set. Let K be any subset of $\mathbb{E}$. We propose the two following selection rules from K to set the value of $f(u)$ in Method 1 in Chapter 3:

$$\begin{aligned} \text{min-rule:} \quad f(u) &:= \min\{k \in K\}; \text{ and} \\ \text{max-rule:} \quad f(u) &:= \max\{k \in K\}. \end{aligned}$$

Let us now provide a precise definition of the rank rules. The intuitive idea of these rules is to remove from the set K the scales higher or lower than a given quantile, which are then considered as non-significant, before applying the min- and max-rules.

Let n be the number of elements in the set K of possible observation scales, *i.e.*, $n = |K|$. Let k be any positive integer less than n. We denote by $\text{rank}_{k/n}(K)$ the element e of K such that there are exactly k distinct elements in K which are less than e. Let p be any real value between 0 and 1, we set $\text{rank}_p(K) = \text{rank}_{\lfloor p.n \rfloor / n}(K)$, where $\lfloor p.n \rfloor$ is the largest integer which is not greater than the real value $p.n$.

Let p be any real value between 0 and 1. We consider the two following additional selection rules from K to set the value of $f(u)$ in Method 1:

$$\begin{aligned} \text{lower p-rank-rule:} \quad f(u) &:= \text{rank}_p(K); \text{ and} \\ \text{upper p-rank-rule:} \quad f(u) &:= \text{rank}_{1-p}(K). \end{aligned}$$

Using the above rules, we now define several strategies which are tested in Section 5.5 to set the value $f(u)$ in Method 1.

When K is equal to the set of all positive observation scales (resp. the set of all positive observation scales not greater than $\overline{\lambda}^{\star}_{\mathcal{H}}(x,y)$) for given edge u and hierarchy $\mathcal{H}$, the result obtained with the min-rule (resp. lower p-rank rule) is called the *min strategy* (resp. *lower p-rank strategy*). With the min-strategy, the value $f(u)$ in Method 1 is set to the minimum scale λ for which the observation criterion holds true. Therefore, the results obtained with this strategy correspond exactly to the results obtained with the method presented in Chapters 3 and 4, and as described by Equation (3.3). The lower p-rank strategy is a new proposal to take more robust decisions, this assertion being practically evaluated in Section 5.5. Intuitively, it can be seen that the lower p-rank strategy considers the lower p percentile of the observation scales between $\lambda^{\star}_{\mathcal{H}}(x,y)$ and $\overline{\lambda}^{\star}_{\mathcal{H}}(x,y)$. Hence, it can be easily seen that, when p is equal to 0, the lower p-rank strategy is exactly the min-strategy.

For our research, we also consider the *max-strategy* (resp., *upper p-rank strategy*), that is the result obtained with the max-rule (resp., upper p-rank rule) when K is the set of the negative observation scales (resp. the set of the negative observation scales not less than $\lambda^{\star}_{\mathcal{H}}(x,y)$). It can be observed that when the observation criterion is increasing, the min-, max-, lower-p-rank, and upper-p rank strategies all take the same decision. However, as illustrated in Figure 5.2, when the observation criterion is not increasing, these rules provide different alternatives to set up the observation scale $f(u)$ in Method 1.

5.4.2 Filtering the observation scales with connected operators

In the previous section, we proposed several strategies to select an observation scale from the positive and negative observation scales. In particular, in order to provide a more robust alternative to the classical min- and max-rules, the p-rank rule is presented. In this section, following the same goal of robustifying the observation scale selection process, we propose to filter the set of positive and negative observation scales with a family of filters, called connected operators, coming from the field of mathematical morphology. The basic idea of the connected operators is to act at the level of the connected components of a set. Basically, each connected component of the input set is either completely preserved or completely discarded but cannot be partially included in the result of the operator. The choice of the connected components which are kept is based on an attribute designed to measure the importance of the connected components. Such filters are very powerful and able to solve many practical issues in image analysis and signal processing as reviewed by

Salembier and Wilkinson [2009]. In this section, we propose to filter the set of positive and negative observation scales with connected operators before selecting an observation scale with the strategies presented in the previous section. We start this section by defining the notion of a component of a set of scales, and then give a formal definition of a connected operator in this framework. We also describe several measures to assess the importance of a component of scales at the end.

Let K be any subset of $\mathbb{E}$. Any interval $I =]\!]\lambda_1, \lambda_2]\!]$ that is included in K and maximal for this property is called a component of K. The set of all components of K is denoted by $CC(K)$. For instance, if K is the set of all positive (resp. negative) observation scales for a given pair $(\mathcal{H}, u)$, then $CC(K)$ is precisely the set $\Lambda_{\mathcal{H}}(u)$ of all positive observation intervals (resp. the set $\overline{\Lambda}_{\mathcal{H}}(u)$ of negative observation intervals). Any map from the set of all intervals of $\mathbb{E}$ into the set of real numbers is called an *(interval) attribute*. Let μ be any interval attribute, let t be any real number, and let K be a subset of $\mathbb{E}$. We set

$$\gamma_t^{\mu}(K) \quad = \quad \cup\{I \in CC(K) \mid \mu(I) > t\}.$$

When the attribute μ is increasing with respect to its parameter (which can be any interval), the operator γ_t^{μ} is called the *μ-connected opening of parameter t* and is indeed an algebraic opening (*i.e,*, an increasing, anti-extensive and idempotent operator). Hence, using an interval measure or attribute μ, we are able to analyze and decide if a certain observation interval is relevant or not. If an observation interval is not relevant, then, it is simply discarded. Afterward, an observation scale is selected among the remaining scales using one of the strategies presented in the previous section. To decide if an observation interval I is relevant, we consider the following attributes:

$$\text{length:} \quad \mu_1(I) = |I|\,; \tag{5.3}$$

$$\text{depth:} \quad \mu_2(I) = \max\{\mathcal{A}(\lambda) \mid \lambda \in I\}\,; \tag{5.4}$$

$$\text{Ndepth:} \quad \mu_3(I) = \max\{-\mathcal{A}(\lambda) \mid \lambda \in I\}\,; \tag{5.5}$$

$$\text{area:} \quad \mu_4(I) = \sum\{\mathcal{A}(\lambda) \mid \lambda \in I\}\,; \text{ and} \tag{5.6}$$

$$\text{Narea:} \quad \mu_5(I) = \sum\{-\mathcal{A}(\lambda) \mid \lambda \in I\}\,. \tag{5.7}$$

Where $-\mathcal{A}(\lambda)$ refers to the negative observation interval. We remind from Equation (5.1) that $\mathcal{A}(\lambda) = \lambda - D(\mathcal{H}_x^{\lambda}, \mathcal{H}_y^{\lambda})$. An illustration of these attributes is provided in Figure 5.3. The attributes of every positive and negative observation interval can be computed on the fly during the execution of Algorithm 5. In our experiments, the positive observation scales are filtered according to the length (Equation (5.3)), depth (Equation (5.4)) and area (Equation (5.6)) measures, whereas the negative observation scales are filtered according to the length (Equation (5.3)), Ndepth (Equation (5.5)) and Narea (Equation (5.7)) measures. Besides the min-, lower-p-rank, max- and upper-p-rank strategies,

which are presented in Section 5.4.1, we also assess in the next section the results of the min-rule on the result of the μ-connected openings applied to the positive observation scales when the measure μ is either the length, the depth or the area (*i.e.*, either μ_1, μ_2, or μ_4). These three strategies are called the *lower-length, lower-depth and lower-area* selection strategies, respectively. Similarly, we also consider the results of the max-rule on the result the μ-connected openings applied to the negative observation scales, when the measure μ is either the length, the Ndepth or the Narea (*i.e.*, either μ_1, μ_3, or μ_5). These three last strategies are referred to as the *upper-length, upper-Ndepth and upper-Narea* selection strategies, respectively. Overall, in the next section, we assess 10 different strategies to set up the observation scale $f(u)$ at Line 4 of Method 1.

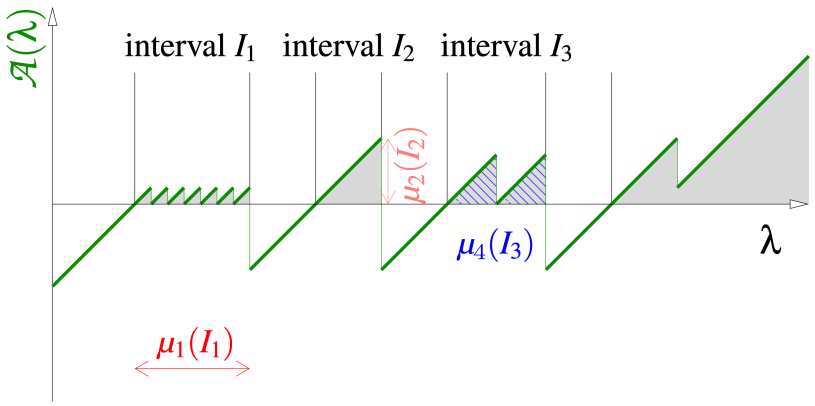


Figure 5.3: Illustration of the attributes used to filter the observation intervals. The observation attribute $\mathcal{A}$ is depicted in green as a function of the scales. The length $\mu_1(I_1)$ of the interval I_1 is represented in red, the depth $\mu_2(I_2)$ of I_2 is represented in pink and the area $\mu_4(I_3)$ of I_3 is shown in blue. Note that, for the sake of readability, the values of the observation attribute $\mathcal{A}$ are depicted for all scales taken in $\mathbb{R}$, whereas the definition is given only for a finite subset $\mathbb{E}$ of $\mathbb{R}$.

5.5 Experiments

This section aims to compare the results of the original HGB method against the results of the new strategies described above. Assessing a hierarchical segmentation, per-se, is a difficult task, as a hierarchical segmentation is generally not an end-goal but an intermediary representation from which the information that is of practical interest for a given task is extracted.

Therefore, we first validate our strategies following quantitative assessments using the framework described by Perret et al. [2018] and that we introduced in Chapter 2. The framework considers images and ground-truth from Pascal VOC 2010 dataset [Everingham

et al., 2010] (2498 images) and Pascal VOC 2012 dataset [Everingham et al., 2012] (3427 objects segmented from the 1449 images).

We also performed qualitative assessments on several images extracted from the Pascal VOC datasets to see how our strategies can lead to better segmentation results.

5.5.1 Quantitative assessments

Let us recall some of the notions about the evaluation of hierarchies that we introduced in Chapter 2. One of the key propositions of Perret et al. [2018] was to assess a hierarchy from different viewpoints corresponding to different use cases of hierarchies. In particular, two aspects are assessed:

- the quality of the "best" cuts or partitions in K regions appearing at the same scale (FHC measure) or at different scales (FOC measure) of the hierarchy. A cut is assessed with respect to a human-provided ground truth and to a segmentation quality measure (Bidirectional Consistency Error). The measure is averaged for every K between 2 and twice the number of regions in the ground truth.

- the ease of finding a set of regions in the hierarchy corresponding to a semantic object when different levels of information are given on the position of the object: an object is considered easy to find when it can be well retrieved with few human-like-provided markers. Here, the retrieved object is a union of regions which, in general, does correspond to any cut of the hierarchy.

The assessment of the first aspect on the Pascal VOC 2010 dataset is summarized with a score called FOC+FHC, whereas the evaluation of the second aspect on the Pascal VOC 2012 dataset is summarized with the ODM score. More details on the computation of these scores are given in [Perret et al., 2018]. For all our experiments, the gradient measure we used was based on colorimetric information from the two pixels of an edge and their Euclidean distance in the RGB color space to produce the edge weight map of the graph representation of the image (see Chapter 2). Furthermore, as recommended by Perret et al. [2018], to get assessable results, we perform an area-simplification post-processing of the hierarchies that remove regions smaller than 0.04% of the image size. For the strategies which depend on a parameter, a search was performed to find the optimal parameter. The tested parameters were:

- $3, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 800, 900, 1000$, and 1050, for the lower-area and upper-narea strategies;

- 3, 5, 7, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, and 70, for the lower-depth and upper-Ndepth strategies;
- 3, 5, 7, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 75, and 80, for the lower-length and upper-length strategies; and
- all values from 0.01 to 0.4 with an increment step of 0.01 for the lower $p-$rank and upper $p-$rank strategies.

The scores obtained with the 10 strategies described in Section 5.4 are presented in Tables 5.1 and 5.2. The associated source code and the saliency maps of our results on numerous images are given at https://cayllahe.github.io/hgbcode.

A visual inspection of the resulting saliency maps shows that there is significant variation between the considered strategies. From the numerical results, we can observe that filtering the observation intervals led to a significant improvement against the raw min- and max-strategies, hence over the original HGB method. More specifically, we observed that:

1. for the best cuts assessment:
   - among the strategies based on min-rule, lower p-rank obtained the highest score (0.9138);
   - among the strategies based on max-rule, upper p-rank obtained the highest score (0.9357);
   - max-rule based strategies systematically performed better than their min-rule counterpart;
2. for the supervised object retrieval assessment:
   - among the strategies based on min-rule, lower-area and lower-depth obtained the highest score (0.877)
   - among the strategies based on max-rule, upper-Ndepth obtains the highest score (0.870);
   - min-rule based strategies systematically performed better than their max-rule counterpart.

From Table 5.1, we can also see that our best FOC score is 0.5194 with the upper-Narea strategy ($parameter = 15$). In Table 5.3, we compare this result to the scores reported by Perret et al. [2018]; our score is higher than the scores obtained by the Watershed-Dynamics RGB hierarchy (see Figure 7 in [Perret et al., 2018]) that achieves 0.359, and we score lower than their hierarchy WS-Area, which obtains 0.529, considering that we are comparable because we are using the same RGB gradient. We can presume

Table 5.1: Best cuts assessment results (Pascal VOC 2010 dataset).

Strategy	Param.	FOC	FHC	FOC+FHC
Min	-	0.4917	0.4010	0.8927
Lower-length	3	0.4904	0.3993	0.8897
Lower-area	5	0.4901	0.3988	0.8889
Lower-depth	3	0.4897	0.3986	0.8883
Lower p-rank	0.17	0.5020	0.4117	**0.9138**
Max	-	0.5184	0.4092	0.9277
Upper-length	35	0.5189	0.4099	0.9288
Upper-Narea	15	0.5194	0.4096	0.9290
Upper-Ndepth	20	0.5189	0.4100	0.9289
Upper p-rank	0.36	0.5193	0.4164	**0.9357**

that our scores could get higher if we use the SED gradient [Dollár and Zitnick, 2015]. Also, Perret et al. [2018] only reported results for FOC scores, which is why we only make comparisons on this metric and we did not consider FHC scores.

Table 5.2: Supervised object retrieval assessment (Pascal VOC 2012 dataset).

Strategy (min-rule)	ODM	Strategy (max-rule)	ODM
Min	0.868	Max-rule	0.865
Lower-length	0.871	Upper-length	0.866
Lower-area	**0.877**	Upper-Narea	0.867
Lower-depth	**0.877**	Upper-Ndepth	**0.870**
Lower p-rank	0.870	Upper p-rank	0.866

From Table 5.2, we can also see that our best ODM scores are obtained by the lower-area and lower-depth strategies, with both getting 0.877. In Table 5.3, we compare this result to the scores reported by Perret et al. [2018]. The ODM score for our Lower-depth strategy is higher than the scores obtained by the best performing unsupervised hierarchy WS-Parents (see Figure 10 in [Perret et al., 2018]) that achieves 0.847. Our strategy performs better than the supervised-based hierarchy UCM-MCG, which scores 0.864; this is particularly interesting since our strategies are totally unsupervised and do not use any learning from a dataset equipped with ground truth. The best-performing supervised-based hierarchy reported by Perret et al. [2018] is COB with 0.919. Nonetheless, we can presume that our scores could get higher if we use more sophisticated gradients such as the SED gradient; still, our results are very promising in the supervised object retrieval assessment proposed by Perret et al. [2018].

5.5.2 Qualitative assessments

For a qualitative assessment, we have extracted three images from the PASCAL VOC dataset (see Figures 5.4(a), 5.5(a), 5.6(a)). We have computed the hierarchical segmentation of the three images using all the presented strategies. The three selected images

Table 5.3: Comparison of results reported by Perret et al. [2018] for FOC-BCE and ODM scores on the Pascal VOC 2010 - Pascal Context and VOC 2012 datasets (see Figures 7, 9 and 10 in [Perret et al., 2018]). Techniques in blue color use learning on datasets, and techniques in black color are unsupervised.

Hierarchies	FOC on VOC 2010	ODM on VOC 2012
COB	**0.722**	**0.919**
UCM-MCG	0.610	0.864
LEP	0.626	0.881
WS-Parents	0.592	0.847
WS Dynamics-RGB	0.359	0.787
WS Area-RGB	0.529	0.770
Our upper-Narea strategy	0.519	0.867
Our lower-depth strategy	0.490	0.877

have the following characteristics: Figure 5.4(a) presents several flat regions, Figure 5.5(a) presents flat regions and several small regions due to texture and Figure 5.6(a) presents non uniform lightning conditions. Then, from each resulting hierarchy, we have extracted an image segmentation and we visually assess such segmentation.

In Figure 5.4, the selected segmentation contains 70 regions. As we can see, the segmentation obtained from the hierarchy resulting from using the min-rule strategy failed to get a good segmentation. There are several coarse regions that do not delineate very well the objects in the image. Between the strategies based on min-rule, lower-depth obtained the best visual result. On the other hand, strategies based on max-rule created coarser regions. Contrary to the ones based in min-rule, such coarse regions delineated better the objects of the image. In particular, for this case, upper p-rank strategy produced the best visual result.

In Figure 5.5, the selected segmentation contains 40 regions. As we can see, the segmentation obtained from the hierarchy resulting from using strategies based on Min-rule strategy obtained a fairly good segmentation of the object. For this case, lower p-rank obtained the best visual result. On the other hand, strategies based on max-rule also tend to create coarser regions. Contrary to the ones based in min-rule. In particular, for this case, upper-Ndepth strategy obtained the best visual result.

In Figure 5.6, the selected segmentation contains 30 regions. As we can see, the segmentation obtained from the hierarchy resulting from using strategies based on min-rule strategy, only the lower p-rank strategy produced a better segmentation of the smaller objects in the image, such as the hands. Strategies based on max-rule also created coarser regions that were not able to capture details of the object, such as the hands.

Overall, we see that there is a clear benefit of filtering the observation scales with the proposed strategies. However, the strategy which leads to the best result depends on the targeted assessment and end-use of the hierarchies.

(a) Original image (b) Min-rule (c) Lower area

(d) Lower P-rank (e) Lower length (f) Lower-Depth

(g) Original image (h) Max-rule (i) Upper Narea

(j) Upper PRank (k) Upper length (l) Upper Ndepth

(m) WS-Area (n) WS-Dynamics (o) WS-Parents

Figure 5.4: Assessment of an image with coarse and flat areas. (b-l) partitions with 70 regions obtained from HGB method with different strategies. (m-o) partitions with 70 regions obtained from hierarchies reported in [Perret et al., 2018]. Strategies based on min-rule produce an over-segmentation. Better results on strategies based on max-rule.

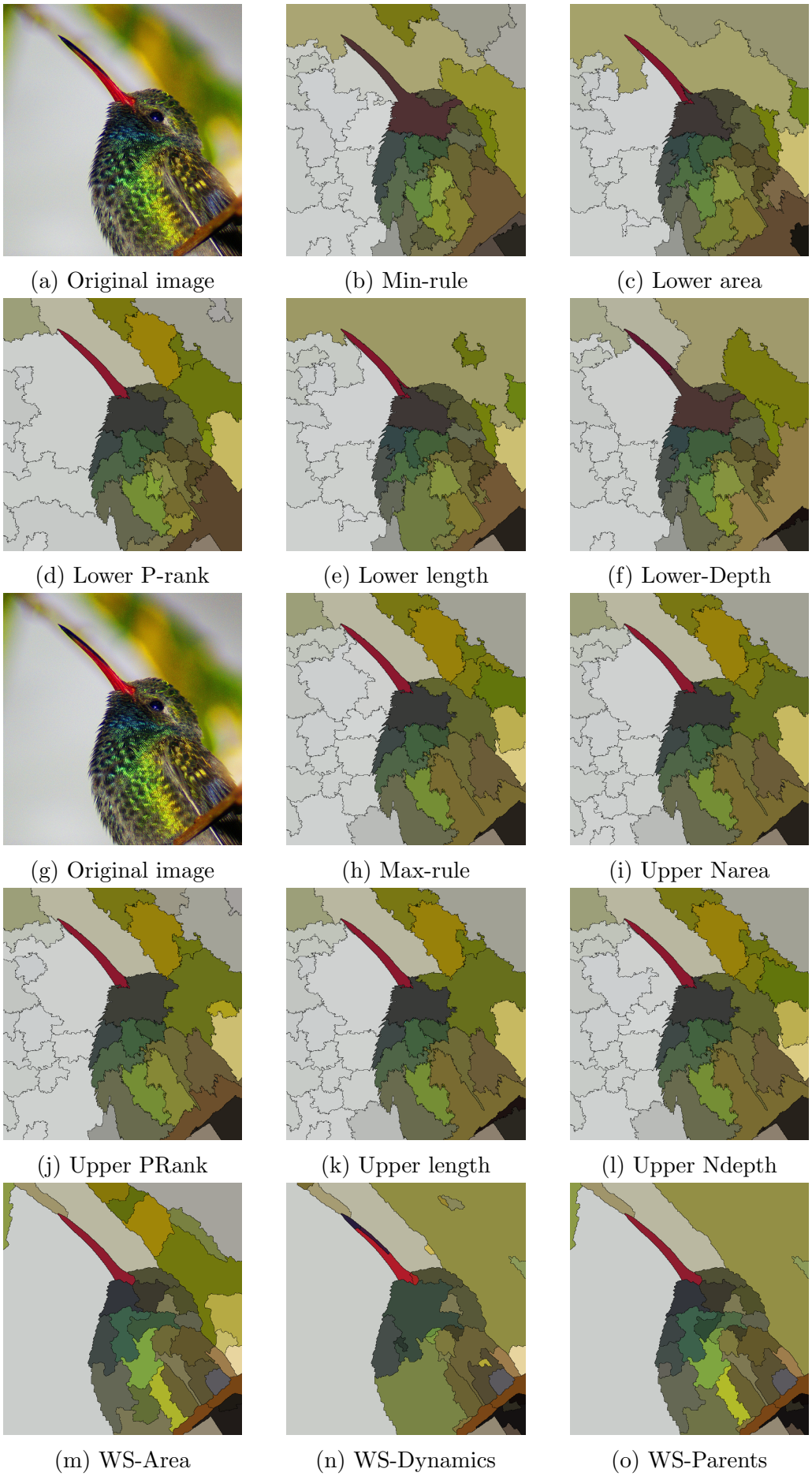
5.6 Concluding remarks

In this chapter, we studied more in depth the minimization equation proposed by the HGB method. We analyzed the non-increasing behavior of the region merging criterion that results from the minimization equation. This analysis allowed us to extend the theoretical properties of such criterion; thus, this chapter defined the notions of edge observation attribute and merging criterion and we were able to establish their non-increasing behavior. We also took advantage of the algorithmic framework introduced in Chapter 4, to propose in this chapter an algorithm to compute all scales for which the criterion is true. This allowed us to efficiently compute all the space of observation scales. Then, by analyzing these scales, we proposed several new strategies to select different observation scales among these values.

Our new strategies were able to improve the quality of the resulting hierarchies compared to the original. Indeed, we assessed the performance of our strategies on Pascal VOC 2010 and 2012 datasets showing that these strategies outperformed the results obtained with original HGB method.

We would like to emphasize that the contributions, we made on this chapter, allow us to fulfill objectives 4 and 5 (see Chapter 1). We have identified a key hyper-parameter; namely, the observation scale selection in the HGB method, and we have proposed new strategies to set up this hyper-parameter (objective 4). Also, by using an specialized framework to evaluate hierarchies, we demonstrated that our new strategies show good performance in a practical evaluation (objective 5).

The new theoretical extensions achieved in this chapter allowed us to understand better the edge observation attributes computed in the HGB method and its criterion. Such criterion is originally formulated by Felzenszwalb and Huttenlocher [2004], it is based on computing the inter- and the within-component difference and measures the relevance of each component to distinguish if there is evidence of a boundary between the two components. In this sense, it is interesting to give more importance to components that follow certain characteristic that a user is more interested in. Indeed, our next chapter does a more in depth analysis of this criterion and we propose new measures that aim to give more relevance to regions that contain certain characteristics.



(a) Original image (b) Min-rule (c) Lower area

(d) Lower P-rank (e) Lower length (f) Lower-Depth

(g) Original image (h) Max-rule (i) Upper Narea

(j) Upper PRank (k) Upper length (l) Upper Ndepth

(m) WS-Area (n) WS-Dynamics (o) WS-Parents

Figure 5.5: Assessments on an image with flat areas and small areas with a lot of textures. (b-l) partitions with 40 regions obtained from HGB method with different strategies. (m-o) partitions with 40 regions obtained from hierarchies reported in [Perret et al., 2018].

(a) Original image (b) Min-rule (c) Lower area

(d) Lower P-rank (e) Lower length (f) Lower-Depth

(g) Original image (h) Max-rule (i) Upper Narea

(j) Upper PRank (k) Upper length (l) Upper Ndepth

(m) WS-Area (n) WS-Dynamics (o) WS-Parents

Figure 5.6: Assessment of an image with non-uniform lighting condition. (b-l) partitions with 30 regions obtained from HGB method with different strategies. (m-o) partitions with 30 regions obtained from hierarchies reported in [Perret et al., 2018]. Strategies

Chapter 6

Region attributes to measure their importance in hierarchical segmentation

6.1 Introduction

In this chapter we present a generalization on the notion of relevance measure originally used by the graph-based method HGB, as well as new relevance measures for the hierarchical graph-based image segmentation. Image segmentation computes a partition by dividing the input image into regions that have some correspondence with the image content. In order to find such correspondence, regions are analyzed according to some regional attribute or characteristic, such as color, intensity, texture, etc.; this makes image segmentation an ill-posed problem, due to the vast diversity and ambiguity of visual patterns in images produced in non-controlled environments. To tackle such a problem, any prior information about the image has to be exploited. Incorporating prior information into a segmentation algorithm is a popular field of research and has been actively developed. In the literature, we find segmentation methods proposing the use of user interaction [Friebel et al., 2022], shape models [Chan and Zhu, 2005, Veksler, 2008], topology [Clough et al., 2019], moments [Felzenszwalb and Huttenlocher, 2004], or adaptations of pre-known models into the segmentation algorithm. Shape models are important if the object has a known geometrical shape, thus it is simpler to model and favor such characteristics in the segmentation. Moments models are popular due to their computational simplicity, *e.g.*, the area of a region.

Let us recall from Chapter 2 that a hierarchical image segmentation computes a series of partitions in a multi-scale manner and its final representation is a hierarchical structure. One of the advantages of hierarchical segmentation is that it benefits the emergence of relevant regions at different scales; *i.e.*, if for a particular problem some region is more relevant, then it should emerge and prevail at different scales in the hierarchical structure. In this chapter, we study a generalization of the criteria used by the HGB method (see Section 3.3) such that new attributes give relevance to regions that contain

some characteristic of interest and should emerge and prevail in the computed hierarchical segmentation.

In this sense, in Section 6.2, we formally introduce a generalization on the region relevance measure that was originally used by the HGB method (see Section 3.3 in Chapter 3). Originally, the HGB method takes into account the area as one indicator to measure the relevance of a region in the hierarchy. The generalization we introduce allows us to consider other non-negative functions for region relevance measure. Thus, in Section 6.3, we introduce a series of functions that compute new regional attributes that aim to measure the relevance of a region based on some characteristic. Specifically, we introduce two relevance measures; one based on the ellipticity of a region (inspired by the popularity of shape models) and the other based on the object saliency of an image. We also present the function that computes region relevance based on area as a reference in this general framework.

We perform qualitative assessments of our new relevance measures in Section 6.4. We can see, from our experiments, that the new relevance measures successfully benefit the emergence of relevant regions based on specific characteristics. Finally, in Section 6.5, we present some concluding remarks from the contributions we made on this chapter and how they relate to the objectives of this thesis.

6.2 Generalizing the region relevance measure

With Equation 3.1, we show the original region relevance measure used by the HGB method. In this equation, we can see that the observation scale of a region R is proportional to its area $|R|$, which means that larger regions are more likely to remain at high observation scales. Thus, from Equation 3.1, we can say that the area of a region is an indicator of the relevance of such region for the observation scale. Consequently, it is natural to generalize Equation 3.1 by replacing the area indicator with a more general relevance measure. In this section, we propose a formalization of this generalization and study other alternatives to the relevance measure based on the area to quantify the relevance of a region.

Indeed, we are able to consider any non-negative function to measure the relevance of a single region considering some characteristics of the region. The goal of proposing a new function is to produce segmentations that allow us to reward/preserve certain regions more than others in the resulting hierarchy. Namely, if we are interested in objects that have a particular characteristic, then we would be interested in giving more relevance to such regions.

In a hierarchical segmentation, partitions of the hierarchy would merge regions that do not contain the desired characteristic at lower levels, and the regions that do contain the desired characteristic would emerge and prevail at higher scales in the hierarchy. In this sense, we can redefine the observation scale of one region relative to another region for hierarchical segmentation depending on relevance measures.

Let γ be any non-negative function that indicates a measure of relevance for each region. Then, the *observation scale of* R_1 *relative to* R_2 (resp. R_2 relative to R_1) in Equation 3.1 is generalized by using γ instead of $|R|$, such that:

$$S_{R_2}(R_1) = (\Delta_{inter}(R_1, R_2) - \Delta_{intra}(R_1))\,\gamma(R_1) \tag{6.1}$$

This generalization can be directly used in Equation 3.2 for computing the HGB method we presented in Chapter 3. We can also see that if $\gamma(R_1) = |R_1|$, then Equation 6.1 allows us to recover exactly the observation scale and observation scale dissimilarity introduced in Chapter 3, namely Equations 3.1 and 3.2 respectively.

The benefit of this generalization is that if for a particular image there is some prior information (*e.g.*, the object of interest has regions with a particular shape), then, by using a non-negative function that computes this attribute for a region, we are now able to favor such regions and make them emerge and prevail in the hierarchical segmentation.

6.3 Non-negative functions for region relevance measure

In this section, we propose a series of non-negative functions that compute regional attributes and whose results can help us measure the relevance of a region R. These functions can be used indistinctively as the function γ for a region R in Equation 6.1.

6.3.1 Function γ_c: region relevance measure based on area

The non-negative function $\gamma_c(R)$ originally proposed by Felzenszwalb and Huttenlocher [2004] measures the area of a region, this regional attribute is computed by counting the number of elements of a region R.

$$\gamma_c(R) = |R|$$

This measure favors the merging of small regions at lower levels of the hierarchy; if there is strong evidence of a boundary between regions they prevail in the hierarchy. The practical results obtained using this relevance measure were assessed in Chapter 5.

6.3.2 Function $\gamma_{\mathcal{E}}$: region relevance measure based on ellipticity

The problem of object detection and recognition in computer vision is still an open problem; one of the approaches to solving such problem is to consider that objects are made of parts that can be captured in elliptical regions. The ellipse is a shape that has been commonly used to represent regions of interest in an image [Zhang and Lu, 2004] and to detect objects in images [Chia et al., 2012]. Indeed, it is a basic shape that has been applied to a wide range of image processing tasks such as identifying objects of approximately elliptic shape such as certain grains, cells, human faces, etc. Thus, it is interesting to study how to capture regions with an elliptical shape and give them more importance in the resulting hierarchy.

The main motivation behind this measure is that by being able to capture such elliptical shapes and under the presumption that objects are made of parts with elliptical shapes. Our resulting hierarchy should be able to capture more naturally the distribution of elliptical shapes of an object from a scale space perspective. It would also be interesting that regions in the resulting hierarchy that do not follow an elliptical shape are vanished or have less importance in comparison to elliptical regions. In order to produce such hierarchy, we propose a region relevance measure that computes the ellipticity of a region and gives more importance to such regions. Flat regions or regions with irregular shapes are not of much interest in this context.

Thus, the regional attribute that this region relevance measure computes is the ellipticity of a region. In order to compute this attribute, there are several techniques in the literature. One effective approach is to rely on moments invariants. We base our ellipticity measure on the work of Rosin [2003], where ellipticity is measured based on the second-order moment (see Section 2 in [Rosin, 2003]). Let $\mathcal{E}(R)$ be a measure that computes the ellipticity of a region R whose output ranges from $[0, 1]$, where 1 is a perfect ellipse.

It is also interesting to be able to add the area of a region to our measure so that we favor regions that are both larger and elliptical. The motivation behind this is that we would like to be able to give importance to regions that are significantly large and elliptical.

In order to integrate the area into our measure, we propose to normalize the area

of a region $|R|$ to a range of $[0, 1]$.

Both normalized functions $\mathcal{E}(R)$ and $|R|$ allow us to measure the ellipticity of a region and its area. It can also be interesting to give more importance to either the ellipticity or its area, this can be achieved by factoring the result of such functions times an integer constant. We combine all these measures into a single non-negative function $\gamma_{\mathcal{E}}(R)$ which is defined as:

$$\gamma_{\mathcal{E}}(R) = \alpha \left(\frac{|R|}{|V|} \right) + \beta \mathcal{E}(R) \tag{6.2}$$

where $|V|$ is the number of vertices of the graph (number of pixels in the image), and α and β are integer constants.

Our proposed function $\gamma_{\mathcal{E}}(R)$ allow us to favor only elliptical regions by setting $\alpha = 0$. If we are interested in both large and elliptical regions, then both α and β must be set correspondingly. Our experiment section contains quantitative assessments of $\gamma_{\mathcal{E}}(R)$ and shows how it can be very flexible to capture both ellipticity and area of a region.

6.3.3 Function γ_s: region relevance measure based on object saliency

We aim to compute an attribute that can help us measure the relevance of a region based on its saliency. This is inspired by the human vision system, which inherently has an attention mechanism to focus on the most salient objects in an image. In computer vision, this approach has translated to a series of algorithms for automatic salient object detection in an image. Our idea is to give priority to regions that belong to salient objects, and such regions should emerge and prevail in the hierarchy while objects that do not belong to visually salient areas in the image are merged at lower levels of the hierarchy.

In order to detect salient objects, some computer vision methods take as input an image and compute a saliency object map (SOM). The resulting SOM assigns a background value for any pixel that do not belong to a salient object (*e.g.*, 0 in case of a gray scale or binary saliency object map), and a value different to the background value to pixels that belong to salient objects.

In order to compute a saliency object map, computer vision methods have relied on handcrafted features and a predefined saliency measure to search for salient pixels [Zhu et al., 2014, Srivatsa and Babu, 2015, Ma et al., 2021]. Deep Convolutional Neural Networks have also been applied to this problem. Some methods perform a patch-wise search to classify pixels or superpixels into salient or non-salient classes [Li and Yu, 2015, Wang et al., 2015, Li and Yu, 2016, Shin et al., 2022]. However, some of them can produce

coarse outputs due to their nature. To address this issue and give finer results, other methods have relied on refinement strategies. For example, Liu and Han [2016] proposed a network that learns global saliency cues and then refines the details of saliency maps. Similarly, Qin et al. [2019] proposed a supervised encoder-decoder network for prediction and a residual refinement module for enhancing the resulting saliency map.

To compute our attribute based on the saliency of an object, we can rely on a function that computes the saliency object map of an image. As we have discussed, there are many algorithms in the literature that can compute such saliency object map; for our proposal, we have chosen to use the method proposed by Qin et al. [2019] as it has been demonstrated to be effective in the context of natural image processing.

Based on the saliency object map SOM_I resulting from the algorithm proposed by Qin et al. [2019], we propose a measure of saliency for a region R. In this sense, let I be an image and SOM_I be the saliency object map of I. We aim to focus only on pixels that belong to salient objects, the idea is to give more importance to such pixels. For every pixel p in I, $SOM_I(p)$ maps a value corresponding to object saliency; thus, a value 0 means that no salient object is identified in the pixel, and any value greater than 0 means that the pixel may correspond to a salient object with a confidence corresponding to the value. We can use the saliency object map to count the number of pixels corresponding to salient objects in a manner similar to the area of a region ($|R|$), see Equation 6.3.

$$Sal_{SOM}(R) = \sum_{p \in R} 1, \{p \in R \mid SOM_I(p) > 0\} \tag{6.3}$$

where R is a region (subset of V).

$Sal_{SOM}(R)$ maps a value corresponding to the object saliency of a region R by counting the number of elements with object saliency greater than 0 for R in the saliency object map. Furthermore, the maximum value for $Sal_{SOM}(R)$ is $|V|$, which means that all pixels in I correspond to salient objects. This can be useful for normalizing this indicator.

Based this notion, we propose $\gamma_s(R)$ as a function to give more importance to such regions. The underlying idea is that salient objects should be prioritized in the resulting hierarchy. Additionally, we can integrate the normalized area of a region into the measure. This approach would allow us to capture regions that are large (relevance based on area) and that do not necessarily belong to the salient object map.

$$\gamma_s(R) = \alpha \left(\frac{|R|}{|V|} \right) + \beta \left(\frac{Sal_{SOM}(R)}{|V|} \right) \tag{6.4}$$

where $|V|$ is the number of vertices of the graph (number of pixels in the image), and α and β are integer constants.

Our proposed function $\gamma_s(R)$ allow us to favor regions that belong to salient objects by setting $\alpha = 0$. If we are interested in a resulting hierarchy that gives more importance to both salient and large regions, then both α and β must be set appropriately.

6.4 Experiments

The goal of this section is to demonstrate the effectiveness of our generalized framework with new relevance measures used in the HGB method. As our result is a hierarchy, assessing such a hierarchy is not a simple task. Here we follow the evaluation framework for hierarchies proposed by Perret et al. [2018], which was presented in Chapter 2, to perform quantitative assessments for hierarchical segmentation. Our tests consider images and their corresponding ground-truth from Pascal VOC 2010 dataset [Everingham et al., 2010], which contains 2498 images. This evaluation framework aims to assess the following two points:

- the quality of the "best" cuts or partitions in K regions appearing at the same scale (FHC measure) or at different scales (FOC measure) of the hierarchy.
- the ease of finding a set of regions in the hierarchy corresponding to a semantic object when different levels of information are given on the position of the object: an object is considered easy to find when it can be well retrieved with few human-like-provided markers.

In order to perform a qualitative assessment of our new relevance measures, we show the saliency maps (SM) of the hierarchies and the segmentations obtained from the hierarchies with a fixed number of regions.

6.4.1 Quantitative assessment of region relevance based on ellipticity

We assessed our techniques with the Pascal VOC 2010 dataset [Everingham et al., 2010] and the FOC+FHC score using the Bidirectional Consistency Error metric (BCE). The framework proposed by Perret et al. [2018] also computes an ODM score (as we have used it in Chapter 5); unfortunately, due to hardware and software constraints, we were not able to compute the latter for this chapter. Consequently, we only show the FOC+FHC score.

Also, as recommended by Perret et al. [2018], we perform an area-simplification post-processing in our resulting hierarchy. In this post-processing step, we remove regions smaller than 0.04% of the size of the image.

For all the experiments of this section, the gradient we use for the graph edge weight is the SED gradient, as recommended by Perret et al. [2018]. We must point out

that this is not the same gradient we used in Chapter 5, where we used the RGB gradient and this can be the reason why we have also obtained better results.

We must note that Pascal VOC 2010 dataset is not a dataset designed specifically for object recognition based on a compactness measure such as ellipticity; the segmentation from these images and their corresponding ground truths are not necessarily elliptical. Nonetheless, due to the generic nature of object recognition based on elliptical shapes, this dataset can still give us information about object segmentation and help us evaluate the function based on ellipticity.

We assessed all the strategies we proposed in Chapter 5 to deal with the non-increasing behavior of the region merging predicate of the HGB method. The segmentation results obtained with these strategies improved the segmentation results obtained with the original HGB method. The result of this first assessment is presented in Table 6.1, In this experiment, we set the parameters $\alpha = 1$, $\beta = 1$ for our function $\gamma_{\mathcal{E}}(R)$; in this setting regions with high ellipticity are given more importance even if they are small (in number of pixels) compared to larger non-elliptical regions. The best performing strategy is upper p-rank with **1.0465**.

Table 6.1: Results for relevance measure based on ellipticity, best cuts assessment results (Pascal VOC 2010 dataset) using $\alpha = 1$, $\beta = 1$.

Relevance measure: Ellipticity				
Strategy	**Param.**	**FOC**	**FHC**	**FOC+FHC**
Min	-	0.5404	0.4507	0.9912
Lower-Length	5000	0.5442	0.4548	0.9990
Lower-area	600	0.5401	0.4508	0.9908
Lower-depth	6000	0.5446	0.4548	0.9994
Lower p-rank	0.25	0.5464	0.4553	1.0016
Max	-	0.5455	0.4558	1.0013
Upper-length	1000	0.5441	0.4550	0.9991
Upper-Narea	5	0.5444	0.4549	0.9994
Upper-Ndepth	6000	0.5434	0.4540	0.9974
Upper p-rank	0.18	0.5719	0.4745	**1.0465**

Our next assessment consists of giving importance to large and elliptical regions. In this sense, we set and evaluate the parameter α for values in $[10, 100, 200, 300, 400, 500, 600, 700, 800, 90$ Our best results were obtained with $\alpha = 600$; Table 6.2 shows the results of this second assessment. The best performing strategy is upper p-rank with **1.0469**. Compared to the results we obtained in Chapter 5 (see Table 5.1), where the best performing strategy got 0.9369, we see an increase in the performance of our strategies using the relevance measure based on ellipticity. Our new formulation for this relevance measure also allowed us to control to which regions we want to give more importance, either large regions based only on their area or we can also give importance to the ellipticity of such regions.

From Tables 6.1 and 6.2, we can also see that our best FOC score is 0.5719 (0.572 rounded to 3 decimals) with the upper p-rank strategy (*parameter* = 0.18). In Table 6.3, we compare our scores to the ones reported by Perret et al. [2018]; we can see that our

Table 6.2: Results for relevance measure based on ellipticity, best cuts assessment results (Pascal VOC 2010 dataset) using $\alpha = 600$, $\beta = 1$.

Relevance measure: Ellipticity				
Strategy	**Param.**	**FOC**	**FHC**	**FOC+FHC**
Min	-	0.5709	0.4746	1.0455
Lower-Length	10000	0.5709	0.4746	1.0455
Lower-area	400	0.5709	0.4746	1.0455
Lower-depth	10000	0.5709	0.4746	1.0455
Lower p-rank	0.3	0.5715	0.4750	1.0464
Max	-	0.5710	0.4746	1.0456
Upper-length	500	0.5709	0.4746	1.0455
Upper-Narea	5	0.5709	0.4746	1.0455
Upper-Ndepth	5000	0.5710	0.4746	1.0456
Upper p-rank	0.12	0.5716	0.4753	**1.0469**

score is higher than the scores obtained by the techniques QFZ and Watershed-Dynamics reported by Perret et al. [2018] which achieved 0.550 and 0.554 respectively; and we scored lower than their best performing hierarchy WS-Parents, which obtained 0.592. Perret et al. [2018] also showed that the best-performing hierarchy is COB with 0.722, however, this is based on a supervised CNN strategy while our techniques are purely non-supervised and still give very practical results.

Table 6.3: Comparison with results reported by Perret et al. [2018] for FOC-BCE scores on the Pascal VOC 2010 dataset - Pascal Context (see Figures 9, 10 in [Perret et al., 2018]). Our results with ellipticity scored higher than other non-supervised techniques and we scored lower than COB which uses learning.

Hierarchies	**FOC-BCE**
QFZ	0.550
WS-Dynamics	0.554
WS-Area	0.588
WS-Volume	0.588
WS-Parents	0.592
COB	**0.722**
Our upper p-rank with ellipticity relevance measure	0.572

6.4.2 Qualitative assessment of the region relevance measure based on Ellipticity

This section shows the qualitative assessment of the region relevance measure based on ellipticity.

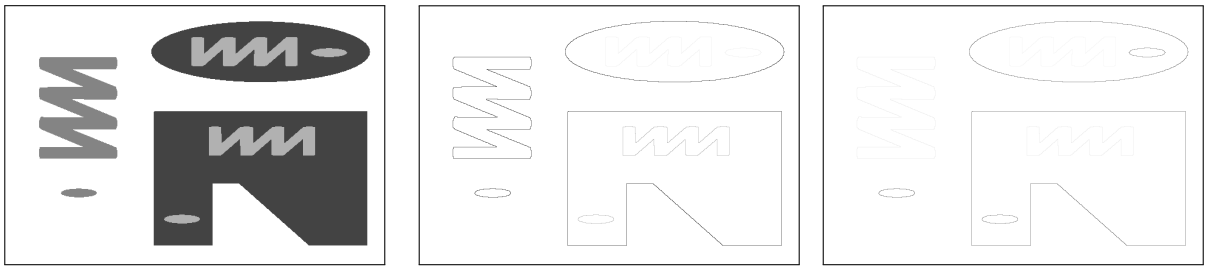
Assessment on synthetic image A first test is performed on a synthetic image as input (see Figure 6.1); the image contains at its left two separated regions, one of which is highly elliptical while the other is more irregular. At the right of the image, there are two nested regions; the region at the top right shows a highly elliptical and large region with two nested regions, among which one is highly elliptical but small and another is a larger region of irregular shape. The bottom right of the image shows a large irregular region with two nested regions, one of which is highly elliptical while another has an irregular shape.

We performed two experiments for our function $\gamma_{\mathcal{E}}(R)$ using the input image: (1) we computed the HGB hierarchy with the original relevance measure by setting the parameters to $\alpha = 1$ and $\beta = 0$; (2) we computed the HGB hierarchy and set the parameters to $\alpha = 0$ and $\beta = 1$. We expect that in the resulting hierarchy from the first experiment, large areas are considered more relevant (as originally proposed by Felzenszwalb and Huttenlocher [2004]) and therefore prevail at higher levels of the hierarchy. To visually confirm this, the saliency map of the resulting hierarchy should show strong borders for large regions. From the second experiment, we expect to see that regions with high ellipticity prevail at higher levels of the hierarchy even if they are smaller than other irregular regions. Figure 6.1 shows the input image and the resulting saliency maps for each computed hierarchy.

From Figure 6.1(c), we can see that the resulting hierarchy with the relevance measure based on ellipticity allow us to give regions with high ellipticity more relevance and make them prevail at higher scales of the hierarchy. Regions with low ellipticity despite being large will banish at lower scales of the hierarchy.

Figure 6.1(b) shows that the hierarchy gives more importance to large regions. We also see in Figure 6.1(b) that the nested structures on the right are given two different scales, despite being the same size and color. The reason for this is that for one, the containing region is larger than the other, see Equation 3.1, where the observation scale of the larger region (larger area) will produce a higher value with respect to the smaller region (smaller area), which makes that the dissimilarity of both regions is governed by the larger region (see Equation 3.2). We see that the containing region is larger than the other, therefore, the observation scale will be higher for the larger region, thus creating the difference we visually see in Figure 6.1(b); we do not see the same behavior in Figure 6.1(c), since the observation scale, in this case, is governed by the ellipticity of the region, not its area.

Assessment on real images Next, we performed a series of experiments on real images, especially where elliptical objects appear. We expect that the relevance measure based on ellipticity gives more importance to elliptical objects. Figure 6.2 shows (a) the input image, (b) the HGB hierarchy using the original relevance measure, (c) the HGB hierarchy



(a) Input image (b) $\alpha = 1$, $\beta = 0$ (c) $\alpha = 0$, $\beta = 1$

Figure 6.1: Test on synthetic images for region relevance measure based on ellipticity.

using the relevance measure based on ellipticity, and (d) the HGB hierarchy favoring both large and elliptical regions. These hierarchies were computed using the gradient measure based on colorimetric information from the adjacent two pixels of an edge and their Euclidean distance in the RGB color space (see Chapter 2).

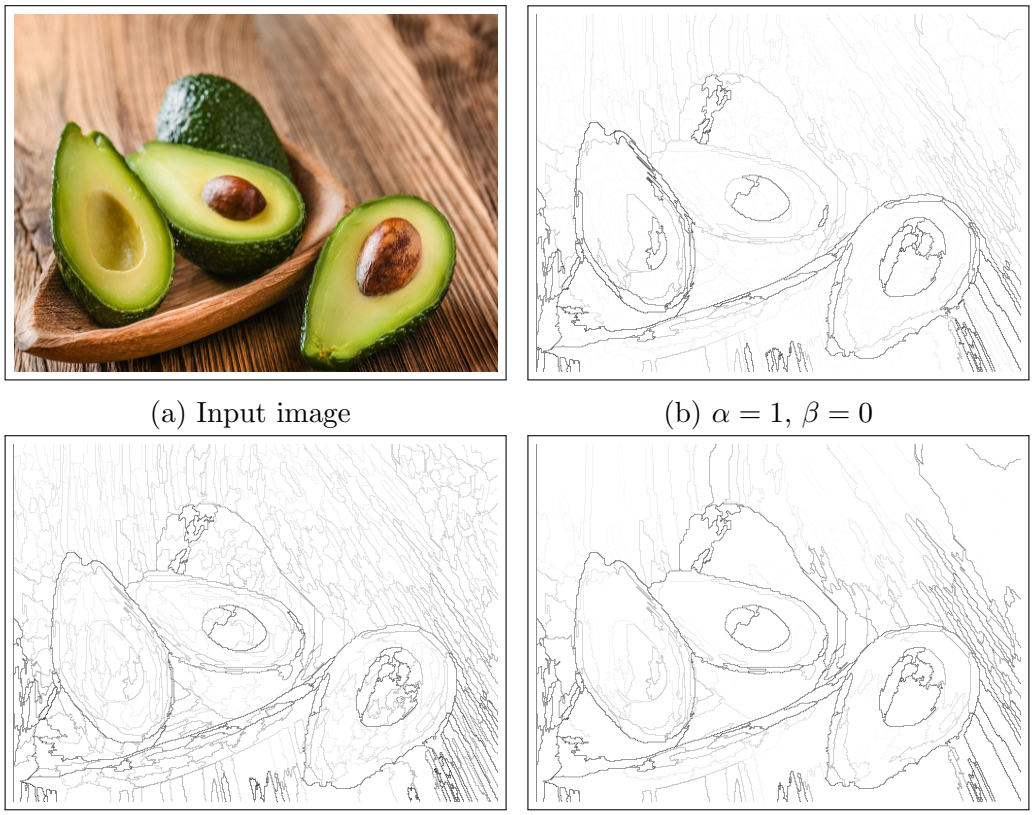
From Figure 6.2, we can see that (b) favors large regions, and the avocados (with high ellipticity) in the middle and in the back will not prevail at higher scales of the hierarchy. In contrast, (c) shows that the new measure gives more relevance to these regions and makes them prevail at higher scales of the hierarchy. However, we can observe in (c) what appears to be an over-segmentation of the background. In (d), we can observe that both elliptical and large regions have more relevance and prevailed at higher scales of the hierarchy, and the over-segmentation present in (c) disappears.

With the same original image of Figure 6.2(a), Figure 6.3 presents a segmentation containing 90 regions at the higher scales of the hierarchy, *i.e.*, the ones that are more relevant. We can visually see how the segmentation result contains more elliptical shapes using our new relevance measure based on ellipticity.

Figure 6.4 shows another assessment of a real image, here we inspect the saliency maps of the hierarchies computed for our relevance measure setting parameters $\alpha \in \{0, 1\}$ and $\beta \in \{0, 1\}$. Figure 6.5 shows the segmentation results obtained from these hierarchies and we also see that the segmentation when $\beta = 1$, contains more elliptical shapes using our new relevance measure based on ellipticity.

6.4.3 Qualitative assessment of region relevance based on object saliency

In order to assess our new region relevance based on object saliency, we selected images that can help us illustrate the effectiveness of this new measure. We set the parameters $\alpha \in \{0, 1\}$ and $\beta \in \{0, 1\}$ for our $\gamma_s(R)$ function. Unfortunately, due to



(a) Input image (b) $\alpha = 1$, $\beta = 0$

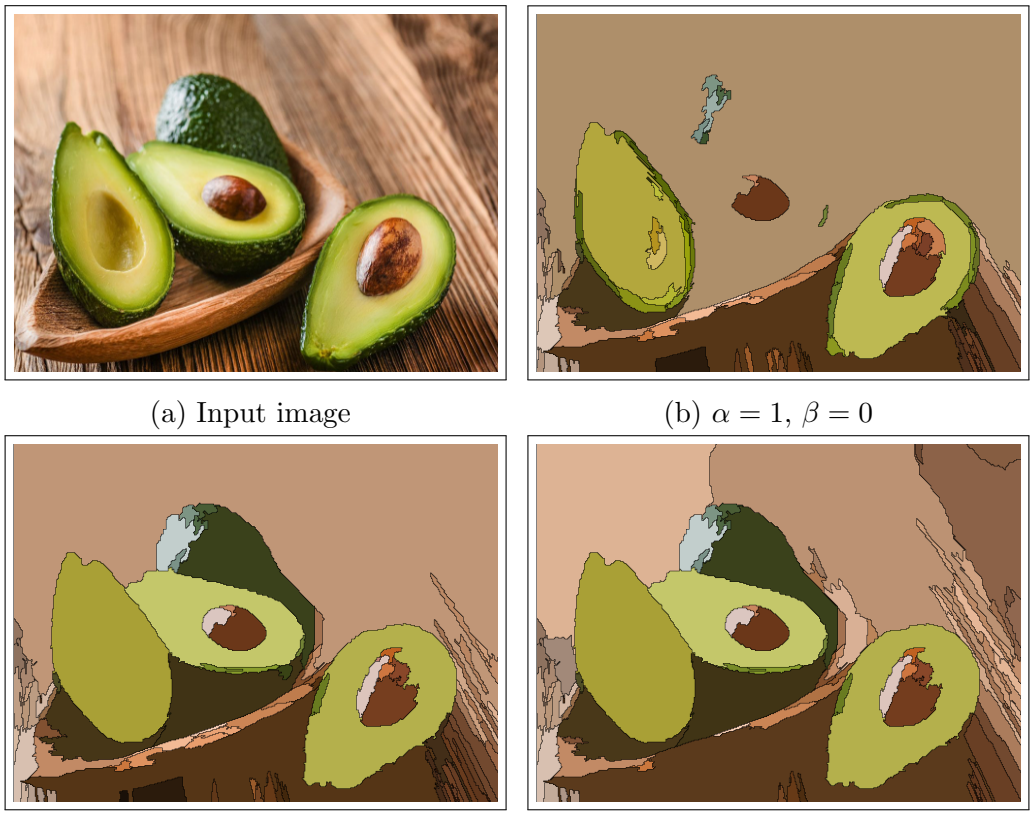
(c) $\alpha = 0$, $\beta = 1$ (d) $\alpha = 1$, $\beta = 1$

Figure 6.2: (b, c, d) show the saliency maps (SM) of the computed HGB hierarchies on the RGB gradient image generated from the input image (a), using the relevance measure based on ellipticity respectively for values $(\alpha, \beta) = (0, 1)$, $(1, 0)$ and $(1, 1)$.

hardware and software constraints, we were not able to compute FOC+FHC and ODM scores for this relevance measure. Therefore, we only present the results of our qualitative assessment.

Our assessments consist of evaluating the resulting hierarchies of two images. For each image, we computed its corresponding object saliency map. We expect to give regions that belong to salient objects more importance compared to other regions in the hierarchy.

In Figure 6.6, (a) shows the input image, which contains an eagle with a background full of textures. (b) shows the object saliency of (a), which corresponds to the eagle. A hierarchical segmentation of the input image would create several small regions on the textured regions. We computed three different hierarchies, (c) shows the segmentation containing 50 regions obtained from the hierarchy computed with parameters $\alpha = 1$ and $\beta = 0$. This resulted in the hierarchy that does not take into account the object saliency of the image, so that we can see that the segmentation contains a lot of regions that belong to the textured background regions. (d) shows the segmentation containing 50 regions obtained from the hierarchy computed with the parameters $\alpha = 0$ and $\beta = 1$. In this case, we see that the backgrounds regions were totally banished in favor of the regions that belong to the salient object. (e) shows the segmentation containing 50 regions obtained

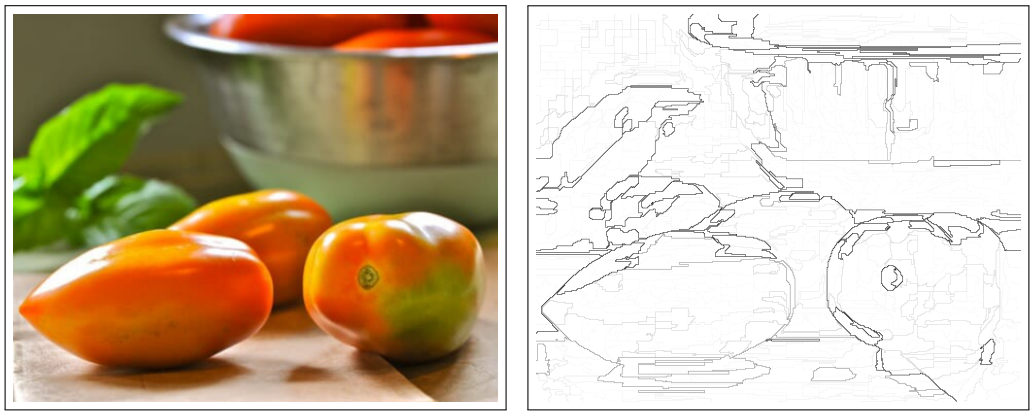


(a) Input image (b) $\alpha = 1$, $\beta = 0$

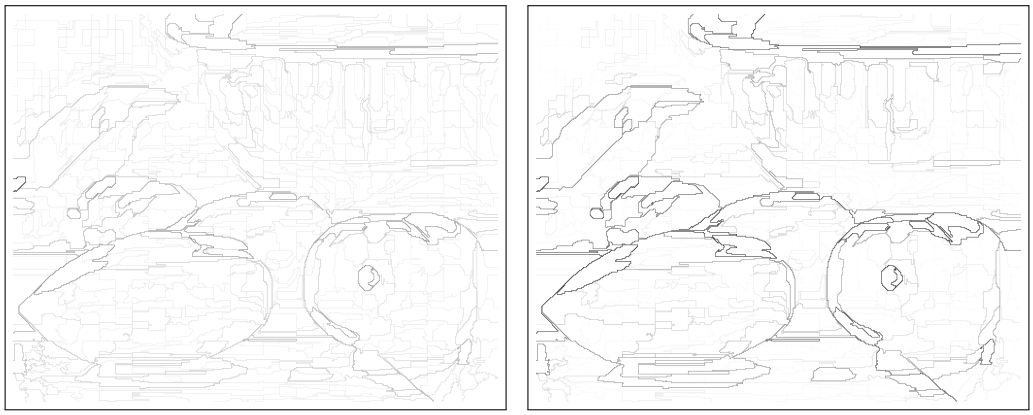
(c) $\alpha = 0$, $\beta = 1$ (d) $\alpha = 1$, $\beta = 1$

Figure 6.3: (b, c, d) show segmentations containing 90 regions from the computed HGB hierarchies on the RGB gradient image generated from the input image (a), using the relevance measure based on ellipticity respectively for values $(\alpha, \beta) = (0, 1)$, $(1, 0)$ and $(1, 1)$.

from the hierarchy computed with the parameters $\alpha = 1$ and $\beta = 1$. This created the hierarchy that gives relevance to large regions that also belong to the salient object. We can see that the segmentation maintains parts of the background but lose some of the details of the eagle (captured in (d)) in favor of larger regions. In Figure 6.7, we also followed the same protocol, (a) shows the input image, which contains a horse with a background full of textures and very strong changes of color and illumination. (b) shows the object saliency of (a), which corresponds to the horse. We computed three different hierarchies, (c) shows the segmentation containing 200 regions obtained from the hierarchy computed with parameters $\alpha = 1$ and $\beta = 0$. This resulted in the hierarchy that does not take into account the object saliency of the image, and consequently many regions from the background appears in the segmentation. (d) shows the segmentation containing 200 regions obtained from the hierarchy computed with the parameters $\alpha = 0$ and $\beta = 1$. In this case, we can see that the backgrounds regions were totally banished in favor of the regions that belong to the salient object, thus more details of the horse are captured in this segmentation. (e) shows the segmentation containing 200 regions obtained from the hierarchy computed with the parameters $\alpha = 1$ and $\beta = 1$. This resulted in the hierarchy



(a) Input image (b) $\alpha = 1$, $\beta = 0$



(c) $\alpha = 0$, $\beta = 1$ (d) $\alpha = 1$, $\beta = 1$

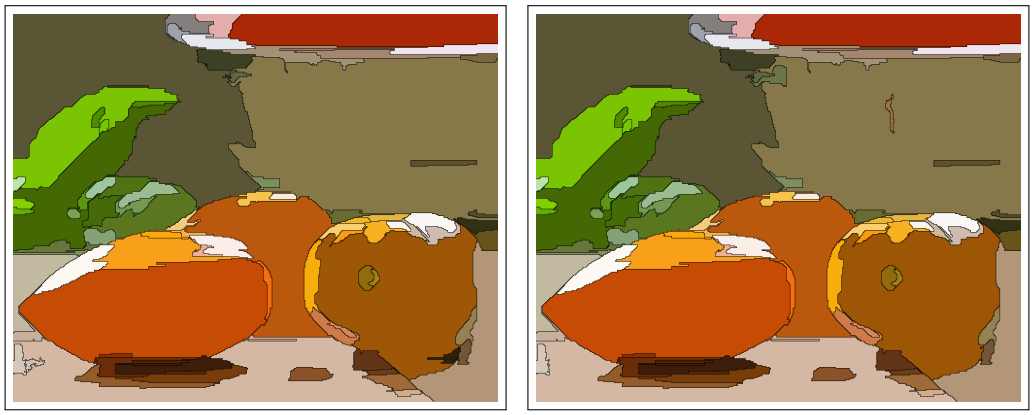
Figure 6.4: (b, c, d) show the saliency maps (SM) of the computed HGB hierarchies on RGB gradient image using the relevance measure based on ellipticity respectively for values $(\alpha, \beta) = (0, 1)$, $(1, 0)$ and $(1, 1)$.

that gives relevance to large regions and also regions that belong to the salient object. Thus we can capture details of both the horse and the background. This means that, although regions that belong to the salient object gained more importance, the hierarchy is still able to preserve regions that do not belong to the salient object and we are still able to produce segmentations of such regions.

6.5 Concluding remarks

In this chapter, we reviewed the merging criterion of the HGB method. The original criterion computes the area of a region to measure its relevance. Then, our motivation was to extend this criterion of the original HGB method and to benefit the emergence of regions with particular characteristics, making them prevail in the final computed hierarchy. To this end, we extended the original formula and presented the generalization of the region

(a) Input image (b) $\alpha = 1, \beta = 0$



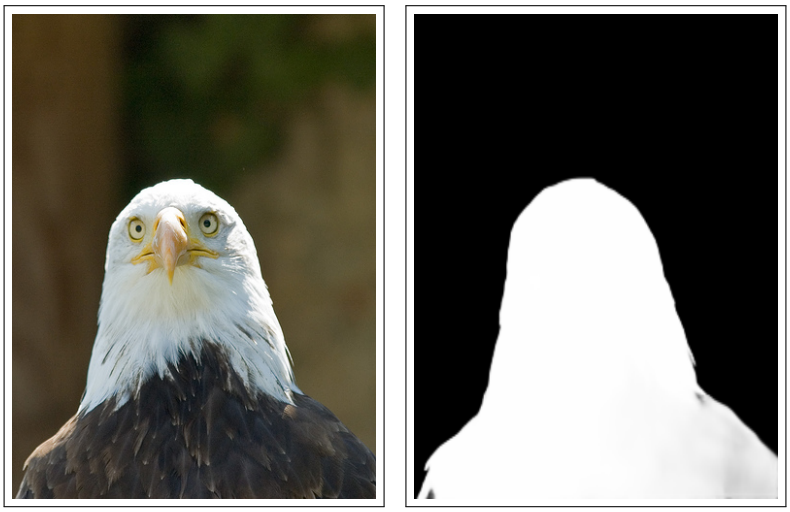
(c) $\alpha = 0, \beta = 1$ (d) $\alpha = 1, \beta = 1$

Figure 6.5: (b, c, d) show segmentations containing 90 regions from the computed HGB hierarchies on RGB gradient image using the relevance measure based on ellipticity respectively for values $(\alpha, \beta) = (0, 1)$, $(1, 0)$ and $(1, 1)$.

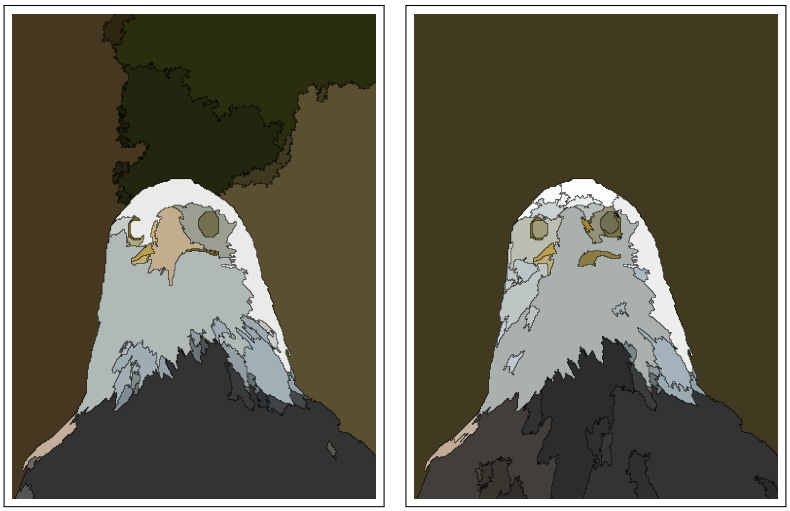
relevance measure. This generalization allowed us to consider new regional attributes. In particular, we considered a set of non-negative functions that compute regional attributes such as area, ellipticity, and saliency of a region to measure the relevance of a region.

Our experiments showed that the new function based on ellipticity is able to perform better than the previous hierarchical segmentation based on the original criterion from the HGB method. Indeed, by following the notion that objects are usually composed of elliptical shapes and regions that have these shapes should be more relevant to others, we were able to obtain better segmentation results, as we can observe from our quantitative assessment. Our new function based on the salient objects of the image was also able to capture and give more details to regions that belong to the salient objects of the image. This can be very handy in situations where we want to capture or retain more details of a certain interest region and ignore other regions that are not so interesting to the user. Our new relevance measure along with the strategies we proposed in Chapter 5 gives us a highly efficient hierarchical segmentation results that are practical for real-world usage.

We would like to emphasize that the contributions we made on this chapter allow us to fulfill objectives 5 and 6 (see Chapter 1). We have generalized the notion of



(a) Input image (b) Object Saliency Map



(c) $\alpha = 1$, $\beta = 0$ (d) $\alpha = 0$, $\beta = 1$

(e) $\alpha = 1$, $\beta = 1$

Figure 6.6: (c, d, e) show segmentations containing 50 regions from the computed HGB hierarchies on RGB gradient image using the relevance measure based on saliency respectively for values $(\alpha, \beta) = (0, 1)$, $(1, 0)$ and $(1, 1)$.

(a) Input image

(b) Object Saliency Map

(c) $\alpha = 1$, $\beta = 0$

(d) $\alpha = 0$, $\beta = 1$

(e) $\alpha = 1$, $\beta = 1$

Figure 6.7: (c, d, e) show segmentations containing 200 regions from the computed HGB hierarchies on RGB gradient image using the relevance measure based on saliency respectively for values $(\alpha, \beta) = (0, 1)$, $(1, 0)$ and $(1, 1)$.

relevance measure originally used by the GB and HGB methods, and we have proposed new relevance measures for the hierarchical graph-based image segmentation (objective 6). Also, by using an specialized framework to evaluate hierarchies, and by performing qualitative assessments, we demonstrated that our new strategies show good performance (objective 5).

Chapter 7

Conclusion

In this chapter, we present a summary of the major contributions of this thesis and review some of our findings. We also discuss some interesting research tasks for further investigation in future work.

7.1 Contributions

The main objective of this dissertation was to study and develop a hierarchical image segmentation method that is practical for real-world usage. We have taken the well established GB method and its hierarchization (see Chapter 2 and 3), and we have established a precise framework to compute the HGB method; then, we have proposed new variations/extensions of the HGB method that allow us to compute a hierarchical segmentation of an image which is efficient and practical for real world usage. Among the contributions of this thesis, we can outline:

- We elaborated a precise formalism to study the formal properties to hierarchize a graph-based segmentation method. As a result of this theoretical background, we presented a general framework from which we were able to propose efficient and exact algorithms to compute a hierarchical image segmentation method. The new framework allowed us to propose a method that, similarly to the extension of watersheds from flat to hierarchical, one level of the resulting hierarchy corresponds to one instance of the flat graph-based segmentation problem.

  Our new framework is focused on the two main steps of the HGB method for improving efficiency:

  - (i) the minimization problem involved in the HGB method, and
  - (ii) the computation of the quasi-flat zone hierarchies.

Concerning (i), our framework allowed us to reduce the search space involved in the minimization problem. We considered two applications of this framework leading to two algorithms which both improved the efficiency, in terms of theoretical time complexity and of practical running times, compared to a naive algorithm for solving the minimization problem. Furthermore, due to the proposed framework, the proof of correctness for each of these two algorithms was provided. Concerning (ii), in order to compute efficiently the quasi-flat zone hierarchy, we considered a non-incremental and an incremental algorithm based on [Najman et al., 2013] and on [Havel et al., 2019], respectively. Even if the worst-case complexities of these two algorithms are comparable, the running times of the HGB method are significantly decreased when the incremental algorithm is used instead of the non-incremental one. Overall, on images from the standard BSDS dataset [Arbelaez et al., 2011], the least efficient strategy that we proposed computed the result of the HGB method in more than four hours whereas the most efficient one took about half a second.

- We proposed new observation scale selection strategies leading to better segmentation results. In order to enhance the segmentation results of the hierarchy, we studied the identification of key hyper-parameters of our hierarchical graph-based method and propose new strategies to set up these hyper-parameters. We were able to propose efficient algorithms for each of our new strategies thanks to our previous work on theoretical background in hierarchizing a graph-based image segmentation method.

  To accomplish this, we introduced the notion of edge observation attribute and criterion, and establish their non-increasing behavior. We formulated this notion as a Boolean observation criterion and proposed new strategies to handle the non-increasing behavior of the observation attribute and criterion behind the HGB method. The essence of our strategies is to filter the set of all the scales for which the Boolean observation criterion holds true, before selecting one scale, with a simple strategy (such as the min-decision rule), among the scales remaining after the filtering. We also introduced other filtering techniques such as rank and connected filters [Salembier and Wilkinson, 2009]. We proposed an algorithm to compute all scales for which the Boolean observation criterion holds true. Our strategies aimed to select different observation scales among the set of all scales. We assessed the performance of our strategies on Pascal VOC 2010 and 2012 datasets showing that these strategies outperformed the original HGB method.

- We proposed a generalization on the notion of relevance measure originally used by the graph-based method proposed by Felzenszwalb and Huttenlocher [2004]. Thanks to our generalization, we were able to include new relevance measures in the hierarchical graph-based image segmentation method. One of the indicators of the original

relevance measure is based on the area of a region; thus, favoring large regions with strong boundaries at the higher scales of the hierarchy. The two new relevance measures we proposed, produced hierarchies that give more importance to regions that contain certain characteristics that a user might be interested in. The first measure is inspired by techniques in object detection and recognition in computer vision, where objects of interest can be captured in elliptical regions. Using this notion, we proposed a relevance measure that gives more importance to elliptical regions in the hierarchy. The second measure is inspired by the notion that the segmentation should contain the most salient objects of the image. In this regard, we used an external algorithm to compute the object saliency of the image and we proposed a new relevance measure that incorporates the object saliency of the input image in order to produce a hierarchy that gives more relevance to regions that belong to salient objects.

- During our research we have implemented several algorithms related to the contributions of this thesis, all the code is available at:
  https://cayllahe.github.io/hgbcode

7.2 Future work

The contributions presented in the thesis form a significant part of the research work carried out. Image segmentation is still an open problem, and the contributions made on this thesis allow us to identify potential research gaps that can be pursued in this field. In this section, we list some of the new research topics and tasks that can be pursued.

7.2.1 Larger study of learned gradients

Many graph based image segmentation methods take as input an image and compute a graph representation from this image (*e.g.*, HGB method). Namely, a pixel adjacency graph representation is computed from the input image (see Chapter 2). In this representation, the edges of the resulting graph link neighboring pixels. In order to weight the edges of the graph, many gradients have been proposed in the literature. In this thesis

and for all of our experiments, we have used two types of gradients:

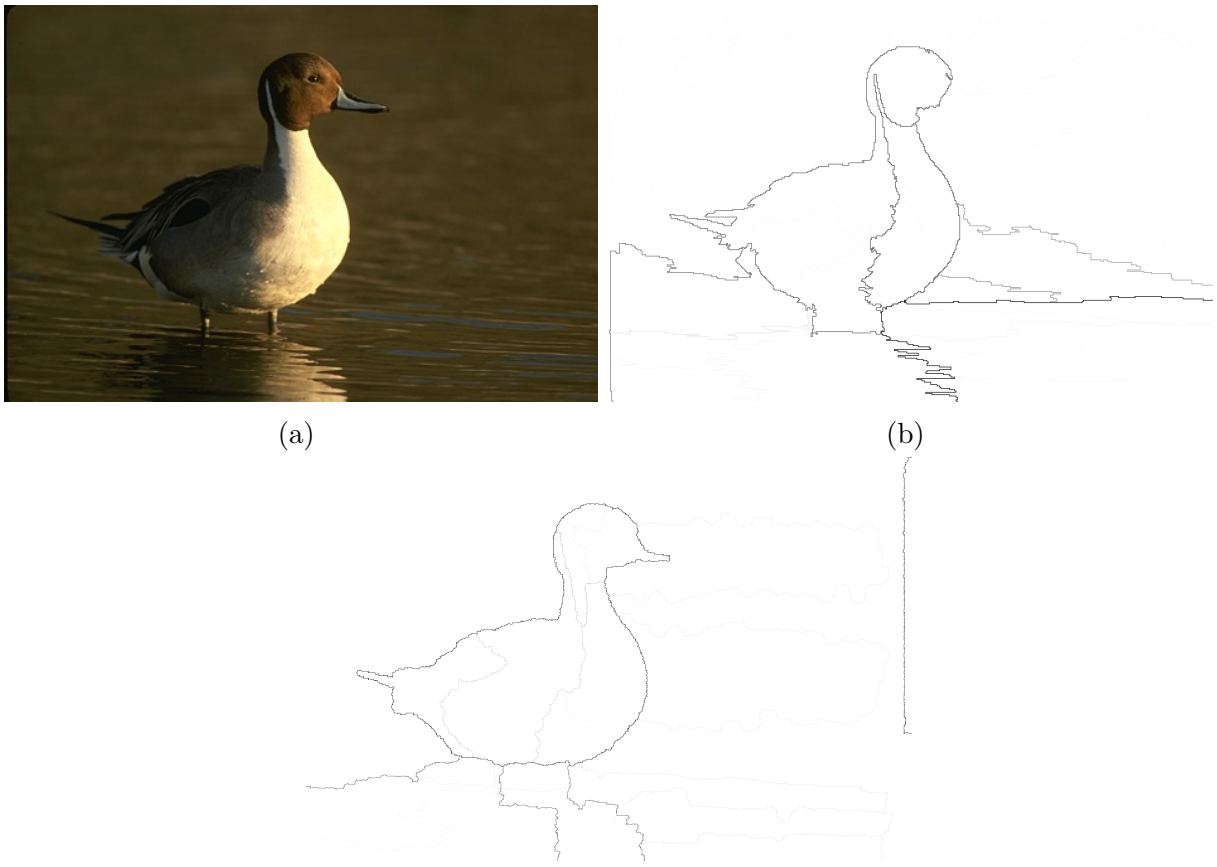
- a gradient based on the colorimetric information, computed from the two pixels of an edge and their Euclidean distance in the RGB color space;
- a supervised gradient estimator called structured edge detector (SED).

The use of learned gradients before the hierarchical construction usually leads to better segmentation results, this was also attested by Perret et al. [2018]. In Figure 7.1, we can observe the saliency maps resulting from the HGB method with upper p-rank selection strategy; Figure 7.1(b) is the resulting hierarchy using the Euclidean distance as gradient; and, (c) is the resulting hierarchy using the structured edge detector from [Dollár and Zitnick, 2015] as gradient. We can see that the supervised SED gradient can help with problems of reflections and brightness present in the image, and the resulting hierarchy can benefit from this. The SED gradient is just one of the many learned gradients we can use to create the pixel adjacency graph representation of the input image. One possible future work would involve a larger study into more recent learned gradients to weight the edges of the graph.

7.2.2 A dissimilarity measure based on learned features

Let us recall that the HGB method involves a minimization problem to determine the scale at which two adjacent regions in a hierarchy should be merged. This minimization problem is solved using a dissimilarity measure that gives an indication of whether or not two regions should be merged or not at a certain scale in the hierarchy. The dissimilarity measure used in the HGB method is a symmetric metric between two regions. If there is evidence of a boundary between two regions at a certain scale, then the two regions should remain separated. To determine the evidence of a boundary between regions, the dissimilarity measure uses information attributes such as the intensity variation and cardinality of a region. While the method uses local cues to determine the evidence of a boundary between regions, the method is still able to capture global characteristics of the image. Currently, neural networks have gained a lot of popularity due to their success in a wide range of applications, including computer vision. In particular, Convolutional Neural Networks (CNN's) can be used to extract both local and global cues from an image. Thus, one possible future work would involve the reformulation of the dissimilarity measure based on local and global cues computed from a neural network.

This new research would involve the study of fusion strategies of the image features extracted from a deep neural network. There are currently a lot of pre-trained networks



(a) (b)

(c)

Figure 7.1: Saliency maps resulting from the HGB method with upper p-rank selection strategy; (a) original image from the BSDS dataset; (b) result using as edge weights Euclidean distance; and, (c) the structured edge detector from [Dollár and Zitnick, 2015].

that can be used as feature extractors (*e.g.*, VGG [Simonyan and Zisserman, 2015], ResNet [He et al., 2016], etc.). In these networks, each middle layer produce features at a certain scale; thus, a fusion strategy would require to incorporate such knowledge into the dissimilarity measure. This can be proposed as a criteria to judge whether or not two regions should be merged or not at a certain scale in the hierarchy.

7.2.3 Improving efficiency in execution time

In this thesis we have proposed efficient algorithm to compute the result of the HGB method. In Chapter 4, we identified two research opportunities: an efficient computation of the minimization problem in the HGB method, an efficient update of the hierarchy at each iteration of the HGB method. This has lead us to propose efficient algorithm to compute the HGB method in Chapter 4, with an average execution time of less than a

second for one image.

Thus, one possible future direction of work is to explore new ideas to improve the efficiency in execution time. One possible approach could be to adapt the HGB method to rely on parallel computation of the hierarchy, such as the work presented by Havel et al. [2019]. Indeed, there is a lot of ongoing research in this area, with some techniques proposing to divide the input image into regular regions and compute the corresponding hierarchy for each region in parallel [Wilkinson et al., 2008]. Then, a region merging strategy can be used to produce the final hierarchy for the whole image. New region merging strategies can be studied and proposed for this purpose.

Improving the efficiency and execution time of the HGB method would allow us to apply our hierarchical segmentation to images with higher resolution, which often require high memory resources [Lefèvre et al., 2022]. Furthermore, a very short execution time would enable us to apply the HGB method to problems that require high-throughput image analysis, such as video segmentation or other real-time processing applications, similar to the technique presented by Gigli et al. [2020]. Consequently, this could be a future contribution that would allow us to apply hierarchical segmentation to a wider range of computer vision problems.

7.2.4 Applying the HGB method to specific problems

In this thesis, we evaluated the performance of the HGB method in the context of natural image analysis. We tested and validated our results using public datasets such as the Berkeley Segmentation Dataset (BSDS) [Arbelaez et al., 2011], Pascal VOC 2010 dataset [Everingham et al., 2010], and the Pascal VOC 2012 dataset [Everingham et al., 2012]. However, the contributions we made in this thesis to the HGB method allow us to study the performance of this method in other contexts of image analysis. For example, our new region relevance measure based on ellipticity can be applied to medical images. In this context, some objects of interest have a particular shape; blood cells and certain types of tissues have an elliptical form. Thus, applying the HGB method with a region relevance measure based on ellipticity seems promising. Additionally, our generalization approach allows us to explore new relevance measures that can be specialized to a specific problem.

In the same direction, video segmentation for specific objects of interest is also a promising future work that could benefit from our new relevance measures. Indeed, many problems in video analysis are interested in specific objects with specific characteristics; *e.g.*, video surveillance, action recognition, etc.

Thus, applying the HGB method to specific problems can be a very interesting path for future research.

7.2.5 Study of the HGB method with other region merging predicates

In this thesis, we started with the general scheme proposed by Guimarães et al. [2012] to hierarchize a graph-based segmentation method. Guimarães et al. [2012] studied the specific case of the image segmentation method proposed by Felzenszwalb and Huttenlocher [2004].

Felzenszwalb and Huttenlocher [2004] proposed a region merging predicate that aims to find evidence of a boundary at a certain scale to determine whether two adjacent regions should be merged or not at that scale. In this thesis, we proposed a theoretical background and efficient algorithms to compute the HGB method using this region merging predicate. However, the HGB method can be used to hierarchize any image segmentation method that utilizes a region merging predicate.

In our future work, we aim to study and explore the HGB method with other region-merging predicates. We have already started on this path with our investigations in Chapter 6, where we generalize the original region merging predicate proposed by Felzenszwalb and Huttenlocher [2004].

References

Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *TPAMI*, pages 898–916, 2011. ISSN 0162-8828.

Pablo A. Arbeláez and Laurent D. Cohen. A metric approach to vector-valued image segmentation. *International Journal of Computer Vision*, 69(1):119–126, Aug 2006. ISSN 1573-1405. doi: 10.1007/s11263-006-6857-5. URL https://doi.org/10.1007/s11263-006-6857-5.

Xue Bai and Guillermo Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007. doi: 10.1109/ICCV.2007.4408931.

Serge Beucher. Watershed, hierarchical segmentation and waterfall algorithm. In *ISMM*, pages 69–76, 1994.

Serge Beucher and Christian Lantuéjoul. Use of watersheds in contour detection. In *International workshop on image processing, real-time edge and motion detection*, 1979.

Petra Bosilj, Sébastien Lefèvre, and Ewa Kijak. Hierarchical image representation simplification driven by region complexity. In Alfredo Petrosino, editor, *Image Analysis and Processing – ICIAP 2013*, pages 562–571, Naples, Italy, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-41181-6.

Petra Bosilj, Ewa Kijak, and Sébastien Lefèvre. Partition and inclusion hierarchies of images: A comprehensive survey. *Journal of Imaging*, 4(2), 2018. ISSN 2313-433X. doi: 10.3390/jimaging4020033. URL https://www.mdpi.com/2313-433X/4/2/33.

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

E. Carlinet and T. Géraud. A comparative review of component tree computation algorithms. *IEEE Transactions on Image Processing*, 23(9):3885–3895, 2014. doi: 10.1109/TIP.2014.2336551.

Edward Cayllahua-Cahuina, Jean Cousty, Silvio Guimarães, Yukiko Kenmochi, Guillermo Cámara-Chávez, and Arnaldo de A. Araújo. A study of observation scales based on

felzenswalb-huttenlocher dissimilarity measure for hierarchical segmentation. In *DGCI*, pages 167–179. Springer, 2019a. ISBN 978-3-030-14085-4.

Edward Cayllahua-Cahuina, Jean Cousty, Yukiko Kenmochi, Arnaldo" de A. Araújo, Guillermo Cámara-Chávez, and Silvio Jamil F. Guimarães. Efficient algorithms for hierarchical graph-based segmentation relying on the Felzenszwalb-Huttenlocher dissimilarity. *IJPRAI*, 33(11):1–28, 2019b.

Edward Cayllahua-Cahuina, Jean Cousty, Silvio Guimarães, Yukiko Kenmochi, Guillermo Cámara-Chávez, and Arnaldo de A. Araújo. Hierarchical segmentation from a non-increasing edge observation attribute. *Pattern Recognition Letters*, 131:105 – 112, 2020. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2019.12.014. URL http://www.sciencedirect.com/science/article/pii/S0167865519303770.

Edward Jorge Yuri Cayllahua Cahuina, Jean Cousty, Yukiko Kenmochi, Arnaldo De Albuquerque Araújo, and Guillermo Cámara-Chávez. Algorithms for hierarchical segmentation based on the Felzenszwalb-Huttenlocher dissimilarity. In *International Conference on Pattern Recognition and Artificial Intelligence*, Montreal, Canada, 2018.

T. Chan and Wei Zhu. Level set based shape prior segmentation. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, volume 2, pages 1164–1170 vol. 2, 2005. doi: 10.1109/CVPR.2005.212.

Y. Chen, D. Dai, J. Pont-Tuset, and L. Van Gool. Scale-aware alignment of hierarchical image segmentation. In *CVPR*, pages 364–372, 2016.

A. Y. Chia, D. Rajan, M. K. Leung, and S. Rahardja. Object recognition by discriminative combinations of line segments, ellipses, and appearance features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1758–1772, Sep. 2012. ISSN 1939-3539. doi: 10.1109/TPAMI.2011.220.

Giovanni Chierchia and Benjamin Perret. Ultrametric fitting by gradient descent. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/b865367fc4c0845c0682bd466e6ebf4c-Paper.pdf.

James R. Clough, Ilkay Oksuz, Nicholas Byrne, Julia A. Schnabel, and Andrew P. King. Explicit topological priors for deep-learning based image segmentation using persistent homology. In *Information Processing in Medical Imaging*, pages 16–28, Cham, 2019. Springer International Publishing. ISBN 978-3-030-20351-1.

Jean Cousty and Laurent Najman. Incremental algorithm for hierarchical minimum spanning forests and saliency of watershed cuts. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 272–283. Springer, 2011.

Jean Cousty, Gilles Bertrand, Laurent Najman, and Michel Couprie. Watershed cuts: Minimum spanning forests and the drop of water principle. *IEEE transactions on pattern analysis and machine intelligence*, 31:1362–74, 09 2009. doi: 10.1109/TPAMI.2008.173.

Jean Cousty, Laurent Najman, and Benjamin Perret. Constructive links between some morphological hierarchies on edge-weighted graphs. In *ISMM*, pages 135–146, 2013.

Jean Cousty, Laurent Najman, Yukiko Kenmochi, and Silvio Guimarães. Hierarchical segmentations with graphs: Quasi-flat zones, minimum spanning trees, and saliency maps. *Journal of Mathematical Imaging and Vision*, 60(4):479–502, 2018. ISSN 1573-7683. doi: 10.1007/s10851-017-0768-7. URL https://doi.org/10.1007/s10851-017-0768-7.

Antonio Criminisi, Toby Sharp, and Andrew Blake. Geos: Geodesic image segmentation. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Computer Vision – ECCV 2008*, pages 99–112, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-88682-2.

H. Digabel and C. Lantuéjoul. Iterative algorithms. In R Verlag, editor, *Proceedings Second European Symposium Quantitative Analysis of Microstructures in Material Science*, pages 85–99, 1978.

Piotr Dollár and Zitnick. Fast edge detection using structured forests. *PAMI*, 37(8): 1558–1570, 2015.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010). *IJCV*, 88(2):303–338, June 2010.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

A.X. Falcao, 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. doi: 10.1109/TPAMI.2004.1261076.

Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, pages 167–181, 2004.

Adrian Friebel, Tim Johann, Dirk Drasdo, and Stefan Hoehme. Guided interactive image segmentation using machine learning and color-based image set clustering. *Bioinformatics*, 38(19):4622–4628, 08 2022. ISSN 1367-4803. doi: 10.1093/bioinformatics/btac547. URL https://doi.org/10.1093/bioinformatics/btac547.

Leonardo Gigli, Santiago Velasco-Forero, and Beatriz Marcotegui. On minimum spanning tree streaming for hierarchical segmentation. *Pattern Recognition Letters*, 138:155–162, 2020. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2020.07.006. URL https://www.sciencedirect.com/science/article/pii/S016786552030252X.

Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., USA, 2006. ISBN 013168728X.

M. Götz, G. Cavallaro, T. Géraud, M. Book, and M. Riedel. Parallel computation of component trees on distributed memory machines. *IEEE Transactions on Parallel and Distributed Systems*, pages 1–1, 2018.

Leo Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, 2006.

Laurent Guigues, Jean Pierre Cocquerez, and Hervé Le Men. Scale-sets image analysis. *IJCV*, pages 289–317, 2006.

Silvio Guimarães, Yukiko Kenmochi, Jean Cousty, Zenilton Patrocinio Jr., and Laurent Najman. Hierarchizing graph-based image segmentation algorithms relying on region dissimilarity - the case of the Felzenszwalb-Huttenlocher method. *Math. Morphol. Theory Appl.*, pages 1–22, 2017.

Silvio Guimarães, Jean Cousty, Yukiko Kenmochi, and Laurent Najman. A hierarchical image segmentation algorithm based on an observation scale. In *SSPR*, pages 116–125, 2012.

Jiří Havel, François Merciol, and Sébastien Lefèvre. Efficient tree construction for multiscale image representation and processing. 16(4):1129–1146, 2019. ISSN 1861-8219. doi: 10.1007/s11554-016-0604-0. URL https://doi.org/10.1007/s11554-016-0604-0.

Yll Haxhimusa, Adrian Ion, and Walter G. Kropatsch. Irregular pyramid segmentations with stochastic graph decimation strategies. In *CIARP*, pages 277–286. Springer Berlin Heidelberg, 2006.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

Josselin Lefèvre, Jean Cousty, Benjamin Perret, and Harold Phelippeau. Join, select, and insert: Efficient out-of-core algorithms for hierarchical segmentation trees. In Étienne Baudrier, Benoît Naegel, Adrien Krähenbühl, and Mohamed Tajine, editors, *Discrete Geometry and Mathematical Morphology*, pages 274–286, Cham, 2022. Springer International Publishing. ISBN 978-3-031-19897-7.

Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5455–5463, 2015. doi: 10.1109/CVPR.2015.7299184.

Guanbin Li and Yizhou Yu. Visual saliency detection based on multiscale deep cnn features. *IEEE Transactions on Image Processing*, 25(11):5012–5024, 2016. doi: 10.1109/TIP.2016.2602079.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. A simple pooling-based design for real-time salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3917–3926, 2019.

Nian Liu and Junwei Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 678–686, 2016. doi: 10.1109/CVPR.2016.80.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. doi: 10.1109/CVPR.2015.7298965.

Wei Ping Ma, Wen Xin Li, Jin Chuan Sun, and Peng Xia Cao. Saliency detection via manifold ranking based on robust foreground. *International Journal of Automation and*

Computing, 18(1):73–84, 2021. ISSN 1751-8520. doi: 10.1007/s11633-020-1246-z. URL https://doi.org/10.1007/s11633-020-1246-z.

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, 2018.

Fernand Meyer. Minimum spanning forests for morphological segmentation. In Jean Serra and Pierre Soille, editors, *Proceedings of the 2nd International Symposium on Mathematical Morphology and Its Applications to Image Processing, ISMM 1994, Fontainebleau, France, September 1994*, volume 2 of *Computational Imaging and Vision*, pages 77–84. Kluwer, 1994. doi: 10.1007/978-94-011-1040-2_11.

Fernand Meyer. The dynamics of minima and contours. In *ISMM*, pages 329–336, 1996.

Fernand Meyer. *Flooding and Segmentation*, pages 189–198. Springer US, Boston, MA, 2000. ISBN 978-0-306-47025-7. doi: 10.1007/0-306-47025-X_21.

Fernand Meyer and Petros Maragos. Morphological scale-space representation with levelings. In *Scale-Space Theories in Computer Vision*, pages 187–198, 1999.

P. Monasse and F. Guichard. Fast computation of a contrast-invariant image representation. *IEEE Transactions on Image Processing*, 9(5):860–872, 2000. ISSN 1057-7149. doi: 10.1109/83.841532. URL https://doi.org/10.1109/83.841532.

Laurent Najman and Jean Cousty. A graph-based mathematical morphology reader. *Pattern Recognition Letters*, 47:3–17, 2014. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2014.05.007. Advances in Mathematical Morphology.

Laurent Najman and Michel Schmitt. Geodesic saliency of watershed contours and hierarchical segmentation. *TPAMI*, pages 1163–1173, 1996.

Laurent Najman, Jean Cousty, and Benjamin Perret. Playing with Kruskal: algorithms for morphological trees in edge-weighted graphs. In *ISMM*, pages 135–146, 2013.

Richard Nock and Frank Nielsen. Statistical region merging. *IEEE Transactions on pattern analysis and machine intelligence*, 26(11):1452–1458, 2004.

Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

Benjamin Perret, Jean Cousty, Olena Tankyevych, Hugues Talbot, and Nicolas Passat. Directed connected operators: asymmetric hierarchies for image filtering and segmentation. *TPAMI*, pages 1162–1176, 2015.

Benjamin Perret, Jean Cousty, Silvio J.F. Guimarães, and Deise S Maia. Evaluation of hierarchical watersheds. *IEEE Transactions on Image Processing*, 27(4):1676–1688, 2018. doi: 10.1109/TIP.2017.2779604. URL https://hal.archives-ouvertes.fr/hal-01430865.

Benjamin Perret, Jean Cousty, Silvio Jamil Ferzoli Guimarães, Yukiko Kenmochi, and Laurent Najman. Removing non-significant regions in hierarchical clustering and segmentation. *Pattern Recognition Letters*, 128:433 – 439, 2019. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2019.10.008. URL http://www.sciencedirect.com/science/article/pii/S0167865519302818.

J. Pont-Tuset, P. Arbeláez, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(1):128–140, 2017.

X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand. Basnet: Boundary-aware salient object detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7471–7481, June 2019. doi: 10.1109/CVPR.2019.00766.

Jimmy Francky Randrianasoa, Camille Kurtz, Éric Desjardin, and Nicolas Passat. Binary partition tree construction from multiple features for image segmentation. *Pattern Recognition*, 84:237 – 250, 2018. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2018.07.003. URL http://www.sciencedirect.com/science/article/pii/S0031320318302358.

Jimmy Francky Randrianasoa, Pierre Cettour-Janet, Camille Kurtz, Eric Desjardin, Pierre Gançarski, Nathalie Bednarek, F. Rousseau, and Nicolas Passat. Supervised quality evaluation of binary partition trees for object segmentation. *Pattern Recognition*, 111:107667, 2021. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2020.107667. URL https://www.sciencedirect.com/science/article/pii/S0031320320304702.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

Christian Ronse. Partial partitions, partial connections and connective segmentation. *Journal of Mathematical Imaging and Vision*, 32(2):97–125, october 2008. ISSN 0924-9907. doi: 10.1007/s10851-008-0090-5. URL https://doi.org/10.1007/s10851-008-0090-5.

Christian Ronse. Ordering partial partitions for image segmentation and filtering: Merging, creating and inflating blocks. *Journal of Mathematical Imaging and Vision*, 49 (1):202–233, 2014. doi: 10.1007/s10851-013-0455-2. URL https://doi.org/10.1007/s10851-013-0455-2.

Paul Rosin. Measuring shape: Ellipticity, rectangularity, and triangularity. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 14, pages 952 – 955 vol.1, 02 2003. doi: 10.1109/ICPR.2000.905609.

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. doi: 10.1109/83.841934.

P. Salembier and S. Liesegang. Ship detection in sar images based on maxtree representation and graph signal processing. *TGRS*, pages 1–16, 2018. ISSN 0196-2892.

Philippe Salembier and Michael HF Wilkinson. Connected operators. *SPM*, 26(6):136–157, 2009.

Philippe Salembier, Albert Oliveras, and Luis Garrido. Antiextensive connected operators for image and sequence processing. *TIP*, pages 555 – 570, 1998.

Jean Serra. A lattice approach to image segmentation. *Journal of Mathematical Imaging and Vision*, 24(1):83–130, 2006. doi: 10.1007/s10851-005-3616-0. URL https://doi.org/10.1007/s10851-005-3616-0.

Gyungin Shin, Samuel Albanie, and Weidi Xie. Unsupervised salient object detection with spectral cluster voting. In *CVPRW*, 2022.

K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. pages 1–14. Computational and Biological Learning Society, 2015.

P. Soille. Constrained connectivity for hierarchical image partitioning and simplification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1132–1145, July 2008. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.70817.

R Sai Srivatsa and R. Venkatesh Babu. Salient object detection via objectness measure. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 4481–4485, 2015. doi: 10.1109/ICIP.2015.7351654.

S. Tanimoto and T. Pavlidis. A hierarchical data structure for picture processing. *Computer Graphics and Image Processing*, 4(2):104–119, 1975. ISSN 0146-664X. doi: https://doi.org/10.1016/S0146-664X(75)80003-7. URL https://www.sciencedirect.com/science/article/pii/S0146664X75800037.

Olga Veksler. Star shape prior for graph-cut image segmentation. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Computer Vision – ECCV 2008*, pages 454–467, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-88690-7.

Luc Vincent and Pierre Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.

Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Deep networks for saliency detection via local estimation and global search. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3183–3192, 2015. doi: 10.1109/CVPR.2015.7298938.

Jan Wassenberg, Wolfgang Middelmann, and Peter Sanders. An efficient parallel algorithm for graph-based image segmentation. In Xiaoyi Jiang and Nicolai Petkov, editors, *Computer Analysis of Images and Patterns*, pages 1003–1010, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-03767-2.

Michael Wilkinson, Hui Gao, Wim Hesselink, Jan Eppo Jonker, and Arnold Meijster. Concurrent computation of attribute filters on shared memory parallel machines. *TPAMI*, pages 1800–1813, 2008.

Yongchao Xu, Thierry Géraud, and Laurent Najman. Connected filtering on tree-based shape-spaces. *IEEE transactions on pattern analysis and machine intelligence*, 38(6): 1126–1140, 2016.

Yongchao Xu, Edwin Carlinet, Thierry Géraud, and Laurent Najman. Hierarchical segmentation using tree-based shape spaces. *IEEE transactions on pattern analysis and machine intelligence*, 39(3):457–469, 2017.

Dengsheng Zhang and Guojun Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1 – 19, 2004. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2003.07.008. URL http://www.sciencedirect.com/science/article/pii/S0031320303002759.

Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2814–2821, 2014. doi: 10.1109/CVPR.2014.360.