Hierarchical image analysis: theory, algorithms, and applications

Benjamin Perret

Habilitation à diriger des recherches

Université Paris-Est

Hierarchical image analysis: theory, algorithms, and applications.

Benjamin Perret
Laboratoire d'Informatique Gaspard-Monge, ESIEE Paris, Université Gustave Eiffel

soutenue publiquement le 11 mai 2021
devant le jury composé de

M. Philippe Salembier, président
Professeur, Universitat Politécnica de Catalunya, BARCELONATECH

M. Jesús Angulo Lopez, rapporteur
Directeur de recherche, Centre de Morphologie Mathématique, Mines ParisTech

Mme Isabelle Bloch, rapporteure
Professeure, Sorbonne Université, Télécom Paris

M. Thierry Géraud, rapporteur
Professeur, EPITA

M. Alexandre Xavier Falcão, examinateur
Professeur, UNICAMP, University of Campinas

M. Laurent Najman, examinateur
Professeur, ESIEE Paris, Université Gustave Eiffel

M. Laurent Wendling, examinateur
Professeur, Université de Paris

Acknowledgments

I would like to warmly thank the members of the jury who accepted to take the time to evaluate my work. Your feedback and our discussions following the presentation were extremely enriching and will continue to stimulate my work.

I would also like to thank all the researchers with whom I have collaborated during these many years: each one has contributed to the work presented in this manuscript. I thank in particular my colleagues of ESIEE Paris who chose to trust me 10 years ago and who helped me to grow as a researcher in a dynamic and friendly environment.

Finally, I would like to thank my family and friends who have given me their unfailing support for so many years.

Contents

Chapter 1

Introduction

This manuscript presents the findings on what I consider to be the backbone of my research activity: hierarchical representations for image analysis. Intuitively, hierarchical image representations come from the very natural idea that the appearance of an image varies according to the distance at which we decide to observe it. From a far distance, we will generally only observe large structures while, closer, we will see that those large structures are themselves composed of finer parts.

I started to develop an interest for this research topic during my PhD thesis which began in 2007 and which was about the automatic characterization of multiband galaxy images. It was a strongly multi-disciplinary subject which led me to explore two very different scientific fields: 1) Monte Carlo Markov chains sampling to solve inverse problems and 2) connected filtering in the framework of mathematical morphology to analyze images. The efficient implementation of such connected filters requires the use of hierarchical image representations called component trees. After the completion of my PhD thesis, I was recruited as a teacher-researcher in computer science at ESIEE Paris and I integrated the A3SI[1] team of the LIGM[2]. Several members of this team already had a long experience with hierarchical representations of images and since then, those representations have become the main theme of my research activity.

The works presented in this manuscript are the results of many local, national and international collaborations, often involving interns and PhD students. My PhD thesis introduced me to develop team working skills in the context of research; the multi-disciplinary nature of the project was materialized by two thesis directors and two advisers. It allowed me to discover the challenge of animating and communicating in a team of researchers with different scientific backgrounds. Since then, I am convinced of the benefit of such collaborations for research: they contribute to the stimulation of the research effort, favor new ideas, and, more importantly, when the domain of competence of the team members are complementary, they support the emergence of novel solutions that would have been impossible to find individually.

These forewords now allow me to give a more precise definition of what I called “my

[1] Algorithmes, architectures, analyse et synthèse d’images

[2] Laboratoire d’Informatique Gaspard-Monge

research activity" at the very beginning of this introduction, by which I mean my constant research effort carried out on selected subjects and in collaborations with several colleagues. The aim of this manuscript is to present the most important results obtained during this research activity. When writing such a document, a balance must be struck between a high-level presentation of the results, which can at times be ambiguous, and a precise mathematical formulation, that can sometimes hide major ideas behind technical details. Here, I have chosen to keep the formalism as reduced as possible to let the reader concentrate on the ideas and I hope that this will make him or her want to read the article cited for a more in-depth presentation.

The main content of this manuscript is divided into three parts. The first two chapters correspond essentially to theoretical, methodological and algorithmic developments in two different paradigms for hierarchical image representations: 1) connected operators and their associated component trees, and 2) image segmentation and their associated hierarchies of segmentations. The third chapter gathers the major applicative developments related to the two first chapters. While I believe that this division into three parts is an important feature of this presentation, it remains somewhat arbitrary given the many links existing between the two paradigms and the fact that applicative and methodological developments are done in parallel and feed each other. The main content of the three chapters is the following:

Connected operators. In this paradigm, an image is viewed as a collection of connected components instead of a set of pixels: we can then decide to preserve or remove each of these connected components. This abstraction level enables to define a large variety of image analysis and processing methods and naturally leads to hierarchical image presentations given by the partial ordering of the connected components of the level sets of the image. In this context, our contributions are organized around three themes:

1. We contributed to the development of the theory of *hyperconnected operators* which is based on a generalization of set connections to lattice of functions. We improved the axiomatic of hyperconnections and we showed how they can be used to devise hyperconnected operators based on associated hierarchical image representations.
2. We proposed a *generic axiomatic framework* for connections and their associated operators. We showed how all the existing theories of connections can be obtained by combining a small number of axioms on top of a very generic theory called inf-structring functions.
3. We generalized the notion of connected operators to *directed connected operators* which can benefit from the rich information given by asymmetric adjacency relations. We studied how the directed connected components of the level sets of an image organize themselves in a hierarchical structure generalizing the hierarchies of connected components. We showed in particular how non local and directed k-nearest neighbor graphs can be used to improve the performances of classical connected filters in bio-medical image analysis.

Hierarchies of segmentations. Those multi-scale image representations intuitively correspond to the decomposition of an image into its objects and the iterative refinement of those objects into their sub-parts. Hierarchies of segmentations are usually used as intermediate representations, offering a reduced search space, in image segmentation or object detection methods. Our main contribution axis in this context are:

1. We studied the theoretical *constructive links* existing between common hierarchical segmentation methods used in mathematical morphology leading to efficient algorithms for computing and for transforming those hierarchies.
2. We performed an in depth study of a particular class of hierarchies of segmentations which corresponds to the solution of a well identified combinatorial optimization problem: *hierarchies of watersheds*. In particular, we provided constructive proofs and efficient algorithms for characterizing, for counting and for transforming hierarchies of watersheds.
3. We proposed a generic method to *optimize hierarchies of segmentations* and hierarchical clusterings according to arbitrary differentiable cost functions using classical gradient descent methods. We showed how some existing NP-hard hierarchical loss functions can be optimized on large datasets in our framework and we proposed novel loss functions taking advantage of the flexibility of the proposed approach.

Assessment and applications. In this chapter we present contributions related to applications.

1. We proposed a novel method to perform a *quantitative assessment of hierarchies of segmentations* based on object detection. This method was integrated in a comprehensive evaluation framework which allowed us to identify a novel attribute to compute hierarchies of watersheds with improved performances compared to classical ones.
2. We developed methods to perform an *automatic characterization of skin aging* in reflectance confocal microscopy images. We proposed image analysis methods based on hierarchical representations to automatically extract aging features from 3d skin images. We showed that these features significantly correlate with features manually extracted by expert dermatologists.
3. We proposed several hierarchical methods to automatically analyze and filter astronomical images. In particular, we developed methods to *detect faint objects in multiband astronomical images* based on hierarchical Markovian models and on statistical testing in hierarchical image representations.
4. We have developed an *open source library HIGRA* for hierarchical graph analysis. This library combines a Python front-end with a C++ back-end and is readily available on major platforms with a single command line. It provides a seamless integration with the rich Python ecosystem for data-analysis and machine learning.

Chapter 2

Connected operators

Spatial relations are a fundamental component of most image processing methods. They can be modeled explicitly, for example when an image is represented as a graph, or implicitly, as in the *sliding window* paradigm used in convolution filters or with structural morphological operators. The study of the various ways of understanding how image pixels are spatially related has led to a large amount of works (see, for example, the review from Braga-Neto and Goutsias [79])

In binary images, the definition of spatial relations between pixels immediately leads to the notion of connected components, *i.e.*, sets of pixels where any two pixels can be joined by a sequence of mutually adjacent pixels. This process can be viewed as a first attempt to perform perceptual grouping. This idea gave rise to the fruitful family of connected operators [102, 225, 232], *i.e.*, operators acting on connected components instead of pixels [220, 226].

Binary connected operators are extended to grey-scale images with the thresholding-stacking approach [132, 150, 254]. Such grey-scale connected operators then have the interesting property to never create or move contours: the only allowed operation is to remove contours. The definition of grey-scale connected operators is deeply linked to hierarchical representations of grey-scale images called component trees: the min/max tree [133, 224] and the tree of shapes [168] (see Figure 2.1). With these hierarchical representations, the construction of a connected operator is generally done in four steps: 1) compute a tree representing the hierarchical decomposition of the image, 2) compute features on the nodes of the tree, 3) filter the tree according to the node features, and 4) reconstruct the filtered tree to obtain a new image.

The operators obtained with this approach are usually referred as *flat* operators because they operate independently on each level set, *i.e.*, flat section, of the input. Those operators became popular for image analysis with applications in medical imaging [181, 255, 266], astronomical imaging [8, 37, 64, 243, 244], vision [152, 265, 267, 268], remote sensing [105, 107, 108, 192, 223] or document imaging [182]. This success can be explained by several reasons. First, efficient algorithms exist for constructing and processing these hierarchical representations [32, 84, 85, 123, 185, 224] and second, the approach is rather intuitive: filters can be designed using rich features dedicated to specific applications. Moreover, those filters benefit from theoretical properties and

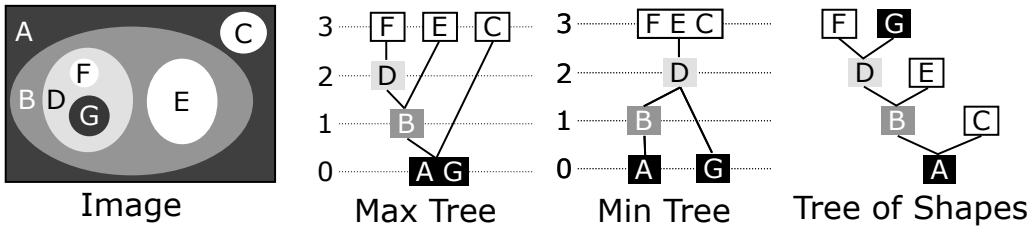


Figure 2.1: An image and its associated max tree, min tree, and tree of shapes. The nodes of the max tree (respectively min tree) are the connected components of the upper (respectively lower) level sets of the image. The nodes of the tree of shapes are the saturated connected components (with their holes filled) of the upper and lower level sets of the image. In all cases, the tree corresponds to the inclusion relation between the nodes. A simple way to design a connected filter is to design a node filtering strategy on such trees.

several classes of connected filters have been defined: flattenings, levellings, or level-set filters [103, 104, 123, 162–164]. The extension of grey-scale connected operators to multivariate images has also been proposed [54, 83, 87, 124, 178, 179, 245, 253].

Our works on connected operators are focused on three axis which are developed in the following sections.

1. In a first sequence of works, we explored the possibility to define grey-scale connected operators without using thresholding-stacking. Actually, one issue with the extension of binary connected operators to the grey-scale case through thresholding-stacking is its sensitivity to *leakage*, *i.e.*, broken contours, and *linkage*, *i.e.*, false contours. The proposed approach is based on the generalization of connections, called hyperconnection, to more general spaces, in particular to the space of functions. In other words, an hyperconnection enables to decompose a function into connected components which are themselves functions. Our works in this direction led to theoretical developments around the theory of hyperconnections which were proven useful in applications in astronomical image processing and document image analysis.
2. The theoretical reflections initiated with our works on hyperconnections led us to propose a general theory that enables to express all known theories of connections and connected operators in a common framework. This work was motivated by the fact that several competing extensions of binary connections had been proposed and the links and differences among them, and among the connected operators they were able to generate, were not well understood. The proposed approach allowed us to identify a set of fundamental properties and to establish bijections between existing connection axiomatics and subsets of those properties.
3. Finally, we explored how directed information can enrich and improve the family of connected operators. We proposed the notion of directed connected components in the framework of directed graphs and we showed how it links with the classical notion of strongly connected components. We studied how, in the grey-scale case, the directed components of a directed graph organize

themselves as a hierarchical representation that generalizes the classical hierarchical representation used to design connected filters. This new representation is not a tree anymore but a directed acyclic graph and we studied how to adapt classical methods and how to define new filtering strategies for this new context. The usefulness of this new directed framework is demonstrated in several applications in medical and biological image analysis.

2.1 Hyperconnected operators

The algebraic notion of connectivity was first defined by Serra [228] and enabled to unify the different notions of connections in graphs and in topological spaces [92, 191, 216, 217]. The original definition was given in the context of the power-set lattice $\mathcal{P}(E)$, with E an arbitrary non empty set. A *set connection* is a family $\mathcal{C}$ included in $\mathcal{P}(E)$ that satisfies three constraints:

C1 - it contains the empty set: $\emptyset \in \mathcal{C}$;

C2 - it contains every singleton of E: $\forall a \in E$, $\{a\} \in \mathcal{C}$; and

C3 - it is conditionally closed under union: the union of a set of intersecting connected elements must be connected: $\forall A \subseteq \mathcal{C}, \bigcap A \neq \emptyset \Rightarrow \bigcup A \in \mathcal{C}$.

The elements of such a family $\mathcal{C}$ are said to be *connected*. An interesting property of $\mathcal{C}$ is that the union of the elements of $\mathcal{C}$ included in a subset A of E and containing a point x of A is connected (*i.e.*, $\bigcup \{C \in \mathcal{C} \,|\, x \in C,\ C \subseteq A\} \in \mathcal{C}$). This element is called *the connected component of A containing x*. Then, the operator that associates the connected component of A containing x to any subset A of E is an opening (it is increasing, anti-extensive and idempotent) called the *the connected opening marked by x*. The connected components of the subset A of E can be equivalently defined as the maximal elements of the set of elements of $\mathcal{C}$ included in A. The set of connected components of a subset A of E forms a partition of A: connected components of A do not overlap and they cover A.

The definition of connections can be immediately extended to complete lattices [53, 76–78, 80, 81, 215, 229, 230]. Intuitively, the idea of this generalization is to define connections where connected components are functions and thus to avoid using thresholding and stacking in the definition of connected operators on general functions: such operators are called *non-flat* connected operators.

Nevertheless, this direct extension of the theory of connections to any complete lattices is hardly applicable in practice as the translation of the property C3 of the set connections into the theory of complete lattices produces an overly strong constraint. The relaxation of the property C3 to accommodate complete lattices has led to the notion of hyperconnections [229] which have been the objects of various theoretical and practical developments [13, 36, 38, 79, 173, 189, 195, 258, 259, 261]. In particular, it has been shown that the approach covers a large variety of morphological operators and concepts including set connected operators (connections are just a special case of the hyperconnections [229]), structural morphology [258] and fuzzy-connectedness [189].

We have contributed to the theory of hyperconnections and its applications in the following works [6, 13, 36, 38], our major contributions are described in the following sections.

2.1.1 Accessible hyperconnections.

In [13, 36], we have identified a sub-class of hyperconnections ensuring that the decompositions into hyperconnected components provide a consistent and intuitive framework for designing hyperconnected filters. In fact, the hyperconnection theory can behave counter-intuitively as it does not guarantee that removing an hyperconnected component from an image will actually change the image.

We proposed a new property that formalizes the following idea: the decomposition of an image in its hyperconnected components must be necessary and sufficient to describe the image. This property is fundamental for image processing as it implies that the decomposition into hyperconnected components completely describes the whole image (sufficient) and that none of its components is useless (necessary). These requirements enforce the consistency of the hyperconnected filters as they ensure that: 1) every deletion of image components will effectively modify the filtered image, and 2) a deleted component can not re-appear in the filtered image. While sufficiency is provided by axioms C1 and C2, we proposed several equivalent formulations of the *necessity* condition that can be used to replace axiom C3. Such hyperconnections were called *accessible* because they also ensure that any hyperconnected component of an element can be obtained by a marked hyperconnected opening.

2.1.2 Hypercomponent tree.

In [13, 38], we have developed a general framework to represent the decomposition of an image into hyperconnected components as an *hypercomponent tree* which corresponds to a generalization of the connected component tree. Such tree is indeed an efficient and intuitive way to design hyperconnected attribute filters or to perform detection tasks based on quantitative attributes: it allows us to reuse the existing algorithms imagined for the connected component trees in this new context. The major difficulty here is that the hyperconnected components of an element are generally not ordered, and thus do not exhibit a tree structure.

We proposed to rely on the notion of z-zones [79] which decompose the umbra of a function into equivalence classes where all points generating the same set of hyperconnected components are said equivalent (see Figure 2.2). We showed that the z-zones of a function can be partially ordered and we gave a sufficient condition to ensure that their Hasse diagram is a tree. If an hyperconnection is *accessible* (see paragraph above), there is a z-zone associated to each hyper-connected component of an element: in other words, in an accessible hyperconnections, the hyperconnected components of a function can be partially ordered through their associated z-zones.

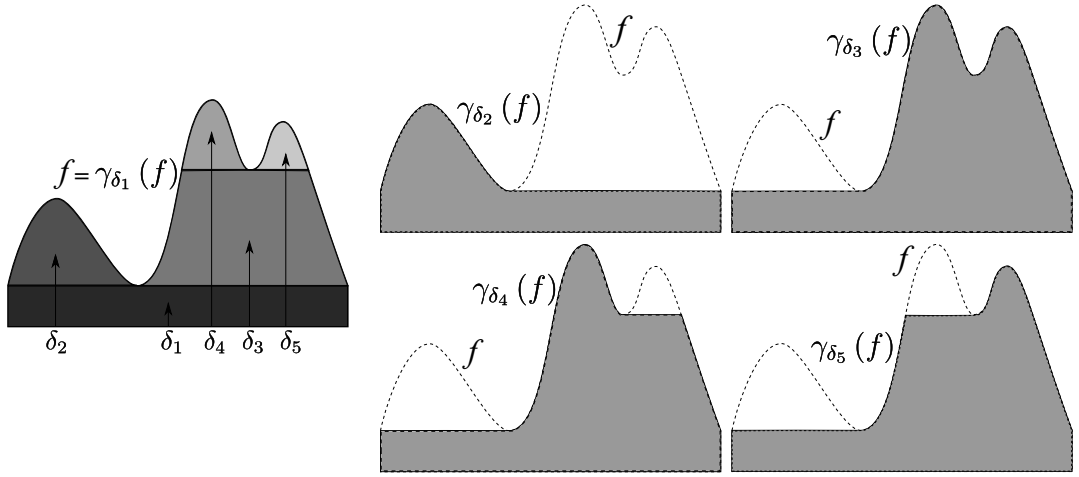


Figure 2.2: Decomposition of a function f with the hyperconnection of functions with a unique maximum [79]. The function f has three hyperconnected components $\gamma_{\delta_2}(f)$, $\gamma_{\delta_4}(f)$, and $\gamma_{\delta_5}(f)$. Left: the function f, five pulses $\delta_1, \ldots, \delta_5$ representatives of the five z-zones of f. The pulse δ_1 (and its equivalence class) is associated to the whole set of hyperconnected components of f and thus the marked hyperconnected opening $\gamma_{\delta_1}(f)$ of f by δ_1 is equal to f. Right: the marked hyperconnected openings of f by $\delta_2, \ldots, \delta_5$.

2.1.3 Hyperconnected filters.

We illustrated the capacity of the proposed hypercomponent tree on two applications [13, 38]. In both cases, the developed hyperconnected operators are based on a fuzzy hyperconnection [189] which is composed of functions with at most one maximum whose dynamics is greater than a given threshold. In other words, a fuzzy hyperconnected function has a single significant maximum and possibly many small local maxima. It is thus expected that the introduction of noise in an image will not lead to the apparition of many new hyperconnected components, as it would be the case with classical connections, but would rather be absorbed by the significant hyperconnected components.

The first proposed application deals with multispectral astronomical image filtering. A classical approach in astronomy for analyzing galaxy images is to fit a 2d brightness profile on the image. The models used for brightness profiles account for the major structures of the galaxy, such as the bulge or the stellar disc, but cannot describe small features such as HII regions (star formation regions) which can have a significant contribution to the total brightness of the galaxy and can thus bias the estimation. Our goal was thus to filter such regions while preserving the rest of the image. Thanks to the properties of the fuzzy hyperconnection and to the proposed hypercomponent tree, a simple extension of the classical area filter to this case provides a good solution to the problem, see Figure 2.3.

The second application deals with document image binarization. Binarization is an important step in document analysis and it requires robust methods able to manage degraded document images of different natures (handwritten, printed) with various scales and varying contrasts. We proposed a novel method which is based on

Figure 2.3: Removal of small features from a galaxy image with an area filtering. The left image is an observation of the galaxy PGC35538 in five bands from near ultra violet to near infrared (only three band are used in the color composition). Then, the first row shows the result of the area filter using from left to right: the hypercomponent tree (objects with an area less than 30 pixels are removed), the max tree (30 pixels) , the max tree (10 pixels). The second row shows the difference between the first row and the original image. This example shows that the proposed approach is able to remove the bright localized structures inside the galaxy which are difficult to model while providing a better preservation of the galaxy morphological features and of the background.

background removal using the hypercomponent tree (see Figure 2.4). The background identification is based on the evolution of the area of the hypercomponent tree nodes compared to their grey level. The proposed method was submitted to the DIBCO 2010 contest (Document Image Binarization Contest) [206] and obtained the 4th place among 16 participants.

2.2 A unifying framework for connections.

In Section 2.1 we have already mentioned the original theory of connections, the extension of connections to general lattices, and their generalization with hyperconnections. Another direction of research in this context is related to the definition of partial connections [210], obtained by dropping the condition C2 on the family of connected elements. As a consequence, with a partial connection, the decomposition of an element into its connected components may contain *holes*: it forms a partial partition. This approach has proven to be useful for the description of iterative processing based on connections, especially in the context of compound segmentation, a theoretical framework for defining and analyzing iterative segmentation methods [194, 211–213, 231].

Another development in this context are attribute space connections [59, 257, 259]

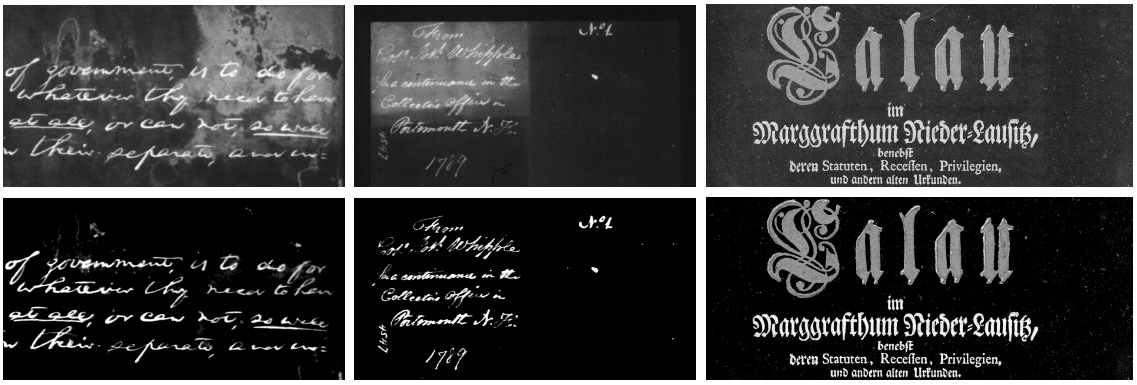


Figure 2.4: Background removal based on hypercomponent tree processing: original document images (top) and corresponding results of the proposed background removal method (bottom).

which are defined on power-set lattices. Their principle is to first plunge the original space into a space of higher dimension, then compute the connected components in this new space, and finally, project them back into the original space. It has been proven that, in the binary case, attribute-space connections generalize hyperconnections [257, 259].

Some of the theory mentioned so far are more general than others or have overlaps, and the situation becomes even more complicated when one considers also the extension of connected filters obtained with thresholding-stacking. This situation is summarized in Figure 2.5.

We have proposed a novel theoretical framework encompassing all previously known connection theory in [6, 41]. The major contributions of these works are described in the following sections.

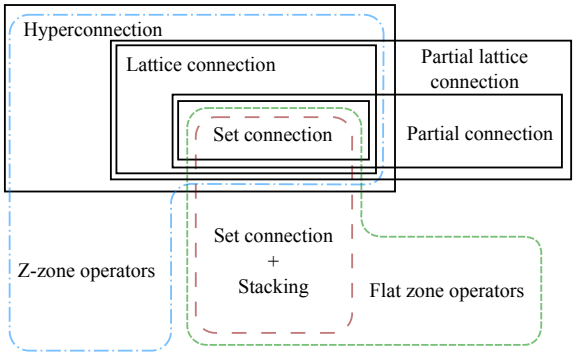


Figure 2.5: Synthetic view of the relations between the different notions of connections and their related connected filters.

2.2.1 Inf-structuring functions

We proposed a general theory for connections [6, 41], based on the new notion of *inf-structuring functions*, that encompasses all previously known approaches to connections in mathematical morphology. This theory does not only have the previous

definitions as special cases but it is also able to directly generate all the connected filters, even those obtained with the stacking technique. Thus, the theory of inf-structuring functions allows us to express the different existing theories in a common framework, giving a better view on their similarities and differences, and easing the transcription of the results obtained in one theory into another one. Moreover, by giving a better view on what is already covered by existing theories we can more easily delimit the *unknown lands*, understand the hypothesis we have to give up in order to start exploring them and avoid redundant work.

The idea of the inf-structuring functions is to start from the least common denominator of all the theories of connections: they all rely on a process that enables to decompose an element of the space into sub-elements. For example, a connection decomposes each element into a partition: *i.e.*, a set of disjoint sub-elements (that cover the element). An hyperconnection decomposes an element into a non-redundant cover: *i.e.*, a set of non-comparable sub-elements that cover the element. A grey-scale connected operator relies on a hierarchy of sub-elements: *i.e.*, a set of sub-elements such that any two sub-elements are either disjoint or comparable.

We call such a mapping that associates each element a of a lattice $\mathcal{L}$ with a set of elements $\mathrm{s}(a)$ of $\mathcal{L}$, such that every element in $\mathrm{s}(a)$ is lower than or equal to a, an *inf-structuring function* [6, 41] (see Figure 2.6).

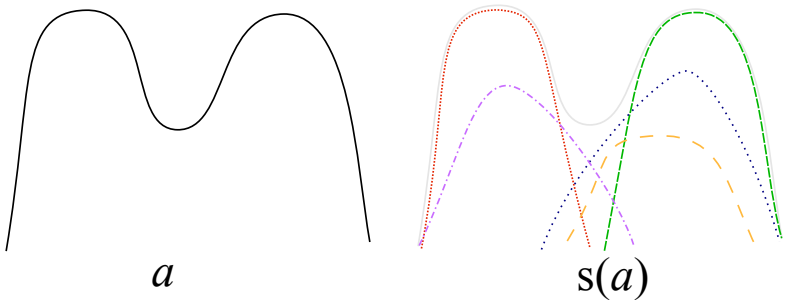


Figure 2.6: Decomposition of a function a into a set of five lower functions $\mathrm{s}(a)$ by the inf-structuring function s.

We identified seven properties [6] characterizing inf-structuring functions. The first three relates to how an inf-structuring function decomposes an element into sub-elements. An inf-structuring function is said:

1. *complete* if the sub-elements are sufficient to recover the original element by supremum.
2. *non-redundant* if no sub-element covers another.
3. *partitioning* if there is no intersection between sub-elements.

The next three characterize how such sub-elements will themselves be decomposed. An inf-structuring function is said:

4. *weakly stable* if a sub-element is always a sub-element of itself.
5. *stable* if a sub-element is always the only sub-element of itself.
6. *strongly stable* if for any set of sub-elements S, the sub-elements of the supremum

of S are exactly S.

And finally, the last one characterizes how the decompositions of two comparable elements relate. An inf-structuring function is said:

7. *increasing* if for any two element a and b such that a is smaller than b, then the sub-elements of a form a refinement of the sub-element of b.

2.2.2 Marked reconstruction operator.

Then, we generalized the notion of marked connected opening to inf-structuring functions: this new operator [6] provides a way to select sub-elements among the sub-elements of an element. In order to ensure that each sub-element of an element can be selected independently, we propose to consider the notion of local minima conditionally to the decomposition. This leads to the definition of the marked reconstruction operator $\beta : \mathcal{L} \times \mathcal{L} \to \mathcal{L}$:

$$\forall a, m \in \mathcal{L},\ \beta(a, m) = \bigvee \min(\uparrow(m) \cap \mathrm{s}\,(a)) \tag{2.1}$$

where a is the processed element, m is the marker and $\uparrow(m)$ is the set of elements that are greater than or equal to m in the lattice $\mathcal{L}$. The reconstruction of a marked by m is thus the supremum of the minima of the family of the upper bounds of m in the family of sub-elements $\mathrm{s}\,(a)$ of a for the inf-structuring function s. An application of β is illustrated in Figure 2.7.

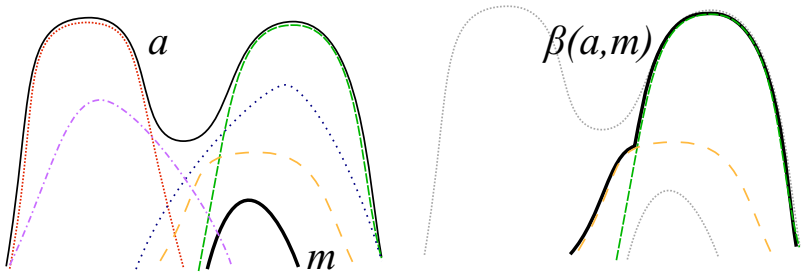


Figure 2.7: Application of the marked reconstruction operator β on the function a decomposed into $\mathrm{s}\,(a)$ (see Figure 2.6) and marked by m. The result $\beta(a, m)$ is equal to the supremum of the dashed green and orange functions which are the smallest sub-elements of a for s greater than m.

We study how the marked reconstruction operator behaves with respect to the identified inf-structuring function properties. In particular we identified which of the seven fundamental properties on the inf-structuring function s implies that the associated marked reconstruction operator β is an opening (proposition 9 of [6]).

2.2.3 Reverse axiomatic of connection theories.

We studied the links between the proposed theory of inf-structuring functions and the existing theories of connections by following the principle of reverse axiomatics: we identified the axioms which are necessary and sufficient to obtain certain properties. For each of the three major theories of connections – connections, hyperconnections

and attributes space connections – and their variations, we identified bijections between inf-structuring functions satisfying a given set set of properties (among the 7 fundamental properties previously proposed) and the connections in the considered theory (theorems 13, 16, and 18 of [6]). A key element in the construction of those bijections was to identify under which conditions an inf-structuring function is associated to a set of *canonical* elements which will correspond to the connected elements of a connection (proposition 8 of [6]). Those equivalence relations are summarized in Table 2.1. It can be seen that for each of the 7 proposed property there is at least one connection theory where it is satisfied and another one where it is not: those properties are thus fundamental to characterize the differences and similarities between the different connection theories.

	Complete	Non-redundant	Partitioning	Weakly stable	Stable	Strongly stable	$\preceq$-increasing
Partial connection [53]		d	X	X	d		X
Connection [229]	X	d	X	X	d		X
Hyperconnection [229]	X	X		X	d		X
Accessible hyperconnection [13]	X	d		d	d	X	X
A-S connection [257]	X						
Strong A-S connection [259]	X			d	X		

Table 2.1: Summary of the equivalence relations between the different theories of connections and the properties of inf-structuring functions. Each line corresponds to one theory of connection. Each column is one of the fundamental properties of an inf-structuring function. For each line, the properties enabling to obtain the equivalence between the connection and the inf-structuring function are marked with an "X". The properties marked by a "d" can be deduced from those marked with an "X": they are necessary but not sufficient.

2.2.4 Characterization of grey-scale connected operators.

We have also studied the case of grey-scale connected operators. We have established that only max-peak operators [195] can be defined directly as (hyper) connected operators. The two other type of grey-scale operators – peak operators [225] and flat-zone operators [225] – cannot be defined in terms of connections without using a thresholding/stacking step. Nevertheless, we have characterized two particular inf-structuring functions that are able to generate all the peak operators and flat-zone operators respectively. The properties satisfied by these inf-structuring functions are summarized in Table 2.2. It can be seen from this, that the introduction of the thresholding/stacking in the construction of these operators weakens the properties of the operators compared to the theory of connection they rely on (see Table 2.1).

	Complete	non-redundant	Partitioning	Weakly stable	Stable	Strongly stable	$\preceq$-increasing
Flat-zone operators	X			X			
Peak operators	X			X			X

Table 2.2: Properties satisfied by the inf-structuring functions that generate flat-zone and peak operators.

2.2.5 Self-dual non-flat flattening.

Based on the proposed notion of inf-structuring function, we have defined a novel extension [41] of a particular class of operators called *flattenings* [162]. Flattenings, which are not to be confused with flat operators, are intuitively operators that reduce the dynamics of a function and that do not introduce new contours. A flattening is said *self-dual* if it processes an image and its opposite symmetrically: it is contravariant with contrast inversions.

The classical definition of the marked flattening [162, 230] is only feasible in sets due to the use of the complementation. We showed that the use of bi-Heyting algebra, *i.e.*, pseudo-complemented lattices (see [142, 209, 238, 239, 247] for details and uses in morphological morphology), removes this limitation and lets us generalize the flattenings in order to use inf-structuring functions in their definitions, *i.e.*, to obtain flattenings based on non flat elements. The construction of the self-dual non-flat flattening operator combines inf-structuring function reconstructions and Heyting algebra operators. We showed that using adapted inf-structuring functions, either based on connections or on hyperconnections, enabled us to recover the original definition of flattenings. Then, we provided, as an example, a simple inf-structuring function whose derived self-dual operator better preserves contrast and does not introduce new pixel values (see Figure 2.8), thus showing the interest of going beyond the stacking/thresholding paradigm in practice.

2.3 Directed connected operators

The various connection theories and their related operators presented in the previous sections all assume symmetric spatial relation: if a pixel x is connected to a pixel y then y is also connected to x. However, several authors have noticed that relaxing this symmetry hypothesis can improve the result of popular image analysis methods such as the min-cuts [75], the random-walkers [234], or the shortest path forests [165]. These works rely on different algorithms based on the directed graph framework, and generally showed an ability to take into account more information than their symmetric counterparts.

Following these successful attempts, we explored in [11] how directed information can enrich and improve the family of connected operators. Note that, while the proposed

Original image f Classical Flattening Non-flat flattening

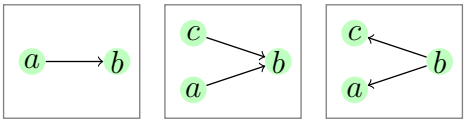
Figure 2.8: Example of application of a self-dual non-flat flattening with inf-structuring functions and bi-Heyting algebra: we can see that with the proposed non-flat flattening the image is smoothed but the highly contrasted area are better preserved compared to the classical flattening (Photo credit: J. Serra).

work is formulated in the framework of directed graph, it has since been proven that the proposed notions can be generalized in an algebraic framework extending the one of connections [214]. The major contributions of this work are described in the following sections.

2.3.1 Directed connected component.

In [11], we introduced the notion of directed connected component: given a vertex x of a directed graph, the *directed connected component of basepoint* x is the set of all vertices which can be reached with a directed path starting from x (see Figure 2.9). This notion generalizes the classical definition of connected components in undirected graphs: when the graph is symmetric, the directed connected component of basepoint x is equal to connected component containing x. However, in general, the set of directed components of a graph do not form a partition of the graph vertices as different directed connected components may have a non empty intersection.

We have proven that the set of directed connected components of a directed graph can be efficiently represented thanks to the classical notion of strongly connected components. Recall that, in a directed graph, a strongly connected component is a maximal set of vertices such that any vertex is reachable from every other vertex in the set. It is well known that this notion induces a directed acyclic graph whose vertex set is equal to the set of strongly connected components of the graph and its edges correspond to the adjacency relations among these strongly connected components. We have established a bijection between the directed components of a graph and its strongly connected components: given two vertices x and y belonging to a same strongly connected components, the directed connected components of basepoints x and y are the same. This bijection implies that the directed acyclic graph of strongly connected components effectively encodes the directed connected components of the underlying graph.



(a) (b) (c)

Figure 2.9: Some elementary directed graphs. The set of directed connected components in (a) (resp. (b) and (c)) is $\{\{a,b\},\{b\}\}$ (resp. $\{\{a,b\},\{c,b\},\{b\}\}$ and $\{\{a\},\{b,a,c\},\{c\}\}$).

2.3.2 Directed components hierarchy.

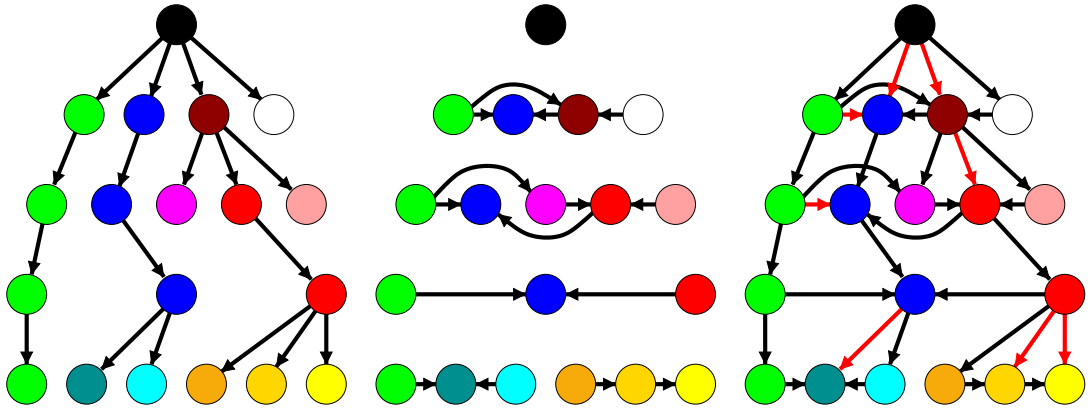
As we have seen, the notion of directed connected component generalizes the one of connected components. This allowed us to define the notion of *directed component hierarchy* which unifies and generalizes all the hierarchical image representations whose definitions are based on connected components [11]: the min/max tree of a vertex weighted graph, the min/max tree of an edge-weighted graph, and the quasi-flat zones hierarchy of an edge-weighted graph (see Section 3.1).

In order to encompass all these different structures, we proposed the notion of a *stack of graphs*, which is defined as a sequence of nested directed graphs. The directed component hierarchy of a given stack of graphs is then defined as the set of directed components of every graph in the stack. Stack of graphs can, for example, be induced by thresholding a vertex weighted graph or an edge weighted graph at all possible values. In such cases and if the base graph is an undirected graph, then the corresponding directed component hierarchy is indeed equivalent to the min or max tree of the original graph.

In the general case, when the graphs in the stack are asymmetric, the directed component hierarchy is no longer a tree: it is a directed acyclic graph with edges representing inter-scale relations (similarly to classical hierarchical tree based representations) but also edges representing the intra-scale relations between strongly connected components (see Figure 2.10). The directed component hierarchy is the key representation to perform directed connected filtering as presented in the following sections.

2.3.3 Algorithm for directed components hierarchy.

In [11] we proposed an efficient algorithm for building the directed component hierarchy of a stack of graphs. The algorithm has a $O(\ell.n)$ time complexity, where n is the size of the graph and where ℓ is the number of levels in the stack. This algorithm is thus well suited for stack of graphs obtained by thresholding graphs with a small number of different weights which is often the case in image analysis (*e.g.*, with pixel values stored in a byte). Recent advances on incremental strongly connected components labeling [63] may enable us to provide other algorithms whose time complexity does not depend of the number of levels in the stack.



(a) (b) (c)

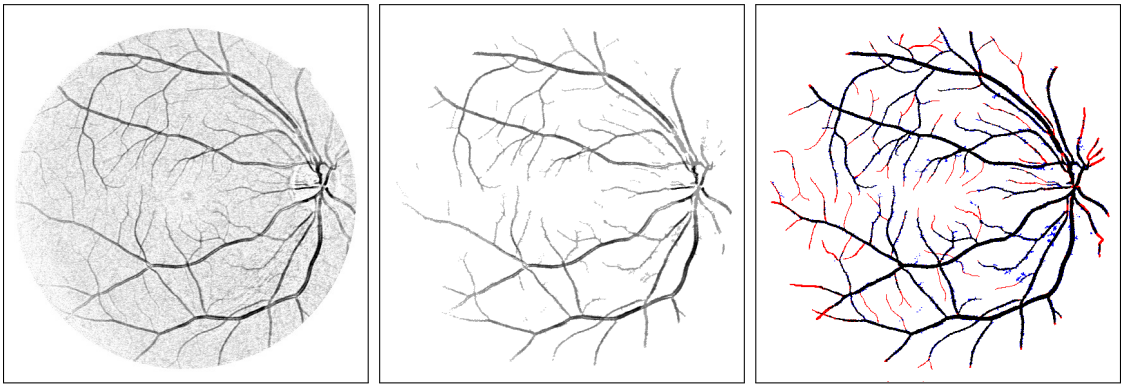
Figure 2.10: Construction of the directed component hierarchy. (a) The strongly connected components associated to a stack of directed graphs generate a tree representing inter-scale relations. (b) The directed acyclic graphs of strongly connected components at each level of the stack of directed graphs encode the directed connected components and represent the intra-scale relations. (c) The directed component hierarchy of the stack of directed graphs is the composition of (a) and (b): it encodes the inter- and the intra-scale relations between the directed connected components (red edges are redundant and can be deduced by transitivity).

2.3.4 Directed connected filtering.

To manage the various cases that may appear when filtering the directed component hierarchy, *i.e.*, a directed acyclic graph, we introduced several strategies [11], called *directed connected filters*. These strategies are designed to ensure the consistency of the node selection process in terms of directed connected components. Thinking in terms of directed connected operators, one may desire to mark each directed connected component as selected or as discarded. However, in contrast to the case of connected operators, we may fall into situations such as the one depicted in Figure 2.9 (b), where two directed connected components overlap. This creates an ambiguous situation if one of them is selected while the other is not selected, hence discarded: one may choose to keep or remove the overlapping components (vertex labeled b in Figure 2.9(b)).

This situation actually generalizes a well known issue arising when one filters a tree with a non increasing Boolean criterion (a criterion that may say to remove a node but not its descendant). The classical solution to deal with a non increasing criterion on trees is to define a regularized criterion according to the min or the max filtering rule [224] which corresponds to the smallest (respectively largest) increasing criterion above (respectively bellow) the base criterion.

With the directed acyclic graph representing the directed component hierarchy, the situation is more complex and it leads to four different regularization strategies resulting of the following choices: 1) do we favor selected or discarded components?, and 2) do we want a smaller or a larger filtering criterion?



(a) Image 19 of DRIVE (b) Filter $I^H_{\sigma_{RV}}$ (c) Segmentation

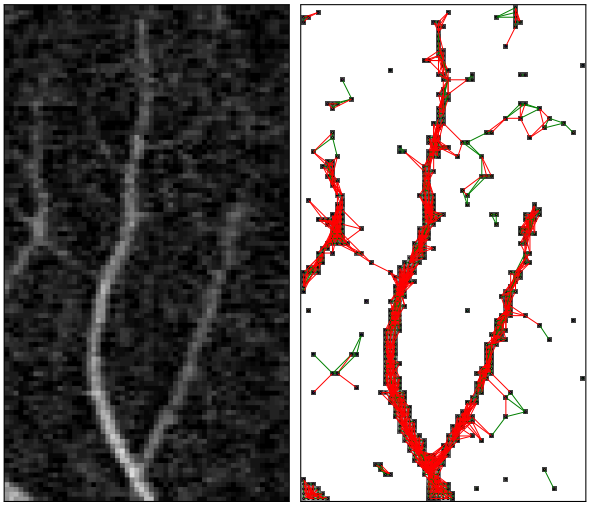
Figure 2.11: Segmentation results on the DRIVE database. From left to right: pre-processed image, filtering result, and evaluation of the segmentation (black pixels are true positives, blues pixels are false positives and red pixels are false negatives).

2.3.5 Image processing with directed connected filters.

We provided several illustrations [11] of the relevance and of the versatility of directed connected filtering for image processing where asymmetric information is taken into account in the form of a directed graph defined over the image pixels.

Eye fundus segmentation. In the first application, the goal is to segment blood vessels in retinal images (see Figure 2.11) in order to help physicians diagnose and follow-up several pathologies of the eye fundus. The difficulty of retinal images lies in the separation of the faint and thin vessels from the background noise. These vessels appear as disconnected groups of pixels that can only be distinguished from the background by their long range spatial coherency. In order to solve this issue, we proposed to construct a non-local directed adjacency relation that allows us to reconnect those groups of pixels, retaining the possibility to reject spurious groups of pixels whose spatial arrangement does not resemble a vessel.

To do so, we proposed to model the image as a semi-local directed graph where each pixel x is adjacent (1) to its four closest spatial neighbors as usually done with the 4-adjacency symmetric relation and (2) to its k brightest neighbors in a window centered on the pixel. This second set of edges is naturally asymmetric: connections are mostly directed from dark to bright pixels (see Figure 2.12). Note that such k-nearest neighbor graph is often used in applications (see, *e.g.*, [118]) without considering its asymmetric aspect (the directed graph is symmetrized by adding its transpose graph). The basic idea behind this construction is to allow directed connected components to *jump* over noisy regions in the faint parts of the blood vessels. In other words, in this directed graphs, blood vessels correspond to the union of the elongated directed connected components rooted in the faint extremities of the vessels. Following this observation, the segmentation mask is then obtained by discarding all directed connected components which are either too small or not elongated enough (see Figure 2.11).



(a) (b)

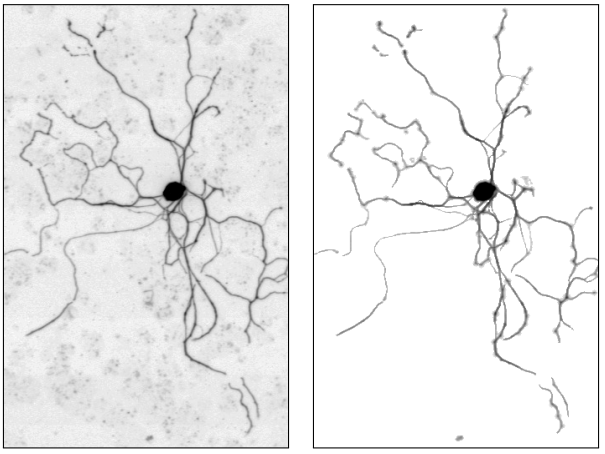
Figure 2.12: Non-local adjacency relation for the directed connected component filtering of blood vessels in retinal images. (a) Distal vessels of the 19th retinal image of the DRIVE database [237]. (b) Adjacency relation shown on a critical threshold of (a). Green links represent symmetric edges while red arrows are asymmetric relations. Each strongly connected component is associated to a color printed in a small circle inside each pixel.

We showed that the proposed method outperformed similar methods based on undirected connected component filters and performed similarly to state-of-the-art methods based on supervised machine learning on the standard DRIVE dataset [237].

Neurite image segmentation. The second application that we proposed is the filtering of neurite images (see Figure 2.13). In this example, we considered a sample image of a neuron grown in vitro with its neurites (*i.e.*, its axon and dendrites). The objective is to derive measures of the neurite tree complexity, which are useful in various toxicology assays, called neurite outgrowth assays [71].

To segment the neurites, we relied on a vesselness-like local object characterization [120], which enables to classify regions into tubes, blobs and background. This allowed us to construct an asymmetric adjacency relation where tubes can be linked to blobs but not the other way around. However, each of the three classes is linked to its own class, while background is linked to all classes. Then, we considered both intensity and geometrical classification in order to filter the image. By imposing asymmetric blob-to-vessel connection, we exploited the fact that the tube classification is under-segmented and we tried to complete the missing information by searching the connections from the blob and the background classes with this more robust tube class.

Myorcardium detection. Finally, the last application considers the integration of an asymmetric a priori knowledge in a marker-based MRI myocardium segmentation procedure. A classical approach to extract a two class *object/background* segmentation



(a) (b)

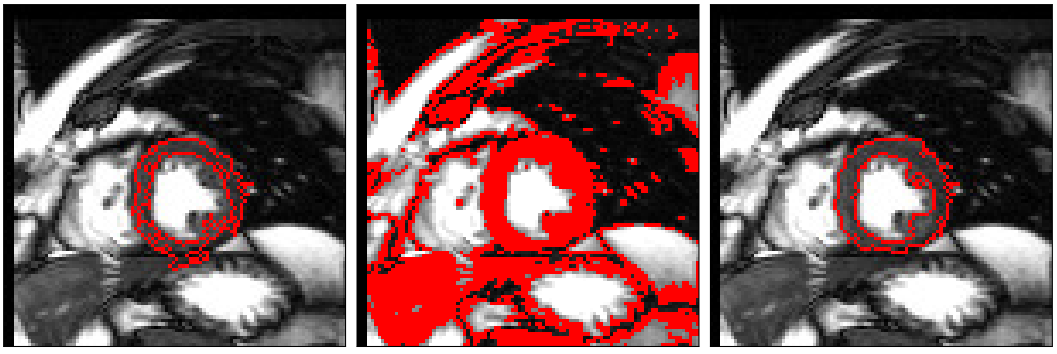
Figure 2.13: (a) Neurite image and (b) Directed connected filtering.

from a hierarchical image representation and markers is to look for the largest components of the hierarchy that have a non empty intersection with the object marker but do not touch the background marker [222]. This procedure can be directly applied to the directed component hierarchy.

Here, we used some a priori knowledge on the intensity distribution of some structures present in the image to define a relevant edge weighted asymmetric graph to represent the image. Actually, in such images, it is known that some extremal intensity pixels are likely to not belong to the myocardium since, in general, they correspond to blood and fat (very bright) or to lungs (very dark): the intuitive idea is then to construct a graph that will ease the connections from the background to these pixels and prevent the object marker to connect to quickly to these pixels. To do this, the edges ending at such pixels are penalized by multiplying their cost by a constant greater than one. Here, the graph is symmetric, but the weights of the edges (x, y) and (y, x) are not equal if one of the vertices x or y is pre-classified as "probably background" while the other is not. Figure 2.14 shows how this strategy can improve the quality of a segmentation compared to the non-directed case (when the weights of the two edges (x, y) and (y, x) are equal).

Moreover, we showed that this marker-based segmentation method defined with the directed component hierarchy is indeed equivalent to the generalization of the image foresting transform [116] to directed graphs presented in [165]. This equivalence derives from the following characterization of the proposed segmentation method: a pixel belongs to the segmented object if the directed min-max distance from this pixel to the object marker is less than the directed min-max distance from this pixel to the background marker.

(a) Original (b) O: myocardium marker (c) B: background marker



(d) Symmetric result (e) S_0: Over-segm. (f) Directed result

Figure 2.14: Marker based segmentation of the myocardium in MRI. (a) Original image. (b) and (c) The object and background markers. (d) Segmentation result obtained with a classical non directed marker based segmentation. (e) Set of pixels that probably contains the myocardium based on a simple brightness prior or, in other words, pixels outside this set are very likely to belong to the background. This information is used to produce asymmetric edge weights. (f) Segmentation result obtained with the asymmetric weights and computed with the directed component hierarchy.

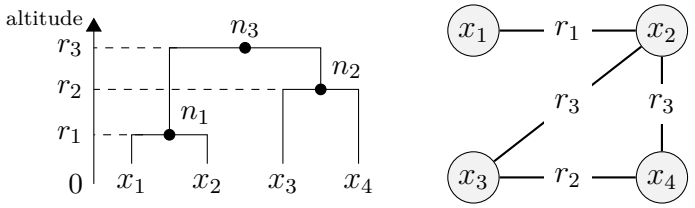
Chapter 3

Hierarchies of segmentations

Image segmentation is the task of grouping the pixels of an image into multiple segments. It is usually seen as an important problem in computer vision that is made particularly difficult by its ill-posed nature. In low level vision for example, one can define an image segment as a connected and homogeneous region of maximal extension. However, in high level vision, a segment can be defined as a semantic object relevant for a given application. In complex scenes, like natural images, it is even possible that the decomposition into semantic objects is itself ambiguous: for example, in a street scene, one could want to segment cars, houses, pedestrians and so on, but one could also want to have a finer level of details where pedestrians are decomposed into their body parts: a head, a trunk, arms, and so on. A natural idea is then to perform a hierarchical segmentation of the image: *i.e.*, to decompose the image into objects and to iteratively refine those objects into parts.

Hierarchies of segmentations were first proposed in the 80's [131,241]. They have since appeared under various names: pyramids, hierarchy of partitions, partition trees, scale-sets. In a hierarchy (of segmentations), an image is represented as a sequence of fine to coarse partitions satisfying the strong causality principle [128, 139, 171]: *i.e.*, any partition is a refinement of the next one in the sequence. They have various applications in image processing and in image analysis: image segmentation [11,58,128, 148,201,208,221,222,265], occlusion boundary detection [130], image simplification [11, 128,236], object detection [222], object proposal [201], visual saliency estimation [270]. In particular, they have gained a large popularity in [58] whose hierarchical approach to the general problem of natural image segmentation outperformed state-of-the-art approaches. Nowadays they are part of state-of-the-art pipelines for image segmentation where they are used as a post-processing of convolutional neural networks [121, 148].

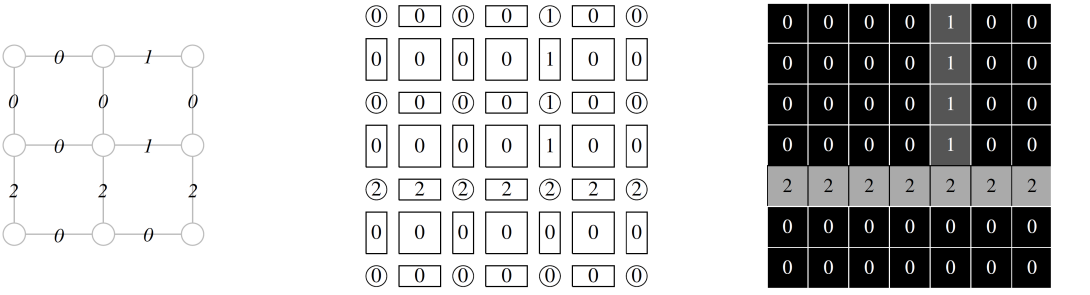
Hierarchical segmentations are usually represented as dendrograms, *i.e.*, trees where the (super)pixels are the leaves and each internal node represents the fusion of its children. A hierarchical segmentation can also be represented as a saliency map [100,184,188], *i.e.*, a characteristic function on the edges of the underlying graph used to model the image domain (see Figure 3.1). One can note that this saliency map can indeed be seen as a restriction of the ultrametric distance associated to the



Dendrogram Saliency map

Figure 3.1: A hierarchy of segmentations represented by a dendrogram (left) and by an equivalent saliency map on a given graph (right). The value of the saliency map on an edge $\{x_i, x_j\}$ is given by the altitude of the lowest common ancestor of x_i and x_j in the dendrogram. For example the lowest common ancestor of x_2 and x_3 is n_3 whose altitude is r_3; the value of the saliency map on the edge $\{x_2, x_3\}$ is thus equal to r_3.

hierarchy to the underlying graph [88, 144]; the two are equivalent if the graph is complete. This duality between the combinatorial representation (the dendrogram) and the functional representation (the saliency map) has been shown useful in image analysis by several authors [55,56,117,137] and will be a key element of the theoretical and algorithmic developments presented in this section. Moreover, saliency maps built on typical 4-adjacency graph used in image analysis have the interesting property of being themselves representable as images in the Khalimsky grid, also called 2d cubical complex [135, 136] (see Figure 3.2). Such representation of a saliency map is sometimes called an ultrametric contour map [58].



Saliency map on a 4-adjacency graph

Equivalent representation on the Khalimsky grid

Visualization as an image

Figure 3.2: The saliency map of a hierarchy of segmentations of an image can be visualized as an image in the Khalimsky grid (source [100]).

Our works on hierarchies of segmentations are focused on three axis which are developed in the following sections.

1. In a first sequence of works, we studied the links existing between some standard hierarchical representations used in image analysis. We showed that several well know hierarchical representations are indeed closely related through constructive relations. Then, we showed how these relations can be translated to efficient algorithms and we proposed a generic framework for hierarchical segmentation construction. This framework is based on an efficient algorithm to construct a *canonical* hierarchical representation that can then be transformed into

other hierarchical representations with linear time post-processings. Finally, we showed how the proposed algorithmic framework can be used to develop efficient filtering methods for hierarchies of segmentations.

2. In a second sequence of works, we focused our study on a particular type of hierarchy of segmentations: the watershed hierarchies. These hierarchies satisfy a scale-wise optimality criterion which enabled us to perform a deep theoretical analysis of these representations. In particular, we addressed the following questions: 1) Can we find a constructive characterization of hierarchical watersheds and thus a recognition algorithm? 2) Can we compute the probability of a given watershed hierarchy and what are the least and most probable watershed hierarchies? 3) Can we transform any hierarchy of segmentations into a *similar* hierarchical watershed ? and 4) Is the set of watershed hierarchies closed by combination ?
3. Finally, we proposed a general approach to hierarchical segmentation cost function optimization with gradient descent. We formulated the optimization problem as a minimization problem on the space of saliency maps. The proposed reformulation led to an unconstrained optimization problem that can be efficiently solved by generic gradient descent methods. The flexibility of our framework allowed us to investigate several new cost functions, following the classic paradigm of combining a data fidelity term with a regularization. While we provide no theoretical guarantee to find the global optimum, the numerical results obtained over a number of synthetic and real datasets demonstrate the good performance of our approach compared to state-of-the-art algorithms.

3.1 Understanding and devising algorithms

The interest of the image processing community towards hierarchical image representations has led to various methods whose links and differences are not always well understood. The mathematical morphology community has in particular focused on the following types of hierarchical representations:

- *Min trees*, *max trees*, and *trees of shapes* [133, 167, 186, 224]: those three hierarchical representations are based on the evolution of the connected components of the level-sets of the pixel values (see Figure 2.1 in Section 2). In those representations, each level of the hierarchy is indeed only a partial segmentation of the pixel set. They are traditionally used to perform connected image filtering and object detection, but they can also be used for general image segmentation, either as contour detectors [265] or as feature extractors [105].
- *Quasi-flat zones hierarchies* [164, 183]: in these hierarchies, the regions of a segmentation at the scale α are defined as the largest connected sets of pixels such that, between any two pixels, there exists a path where the maximal dissimilarity between adjacent pixels is lower or equal than α. When valid paths are further constrained by new criteria, we obtain the family of *constrained connectivity hierarchies* [236].
- *Binary partition trees* [222]: those hierarchies corresponds to the extension

to sparse graphs of classical agglomerative clustering methods developed in data analysis. Starting from the image segmentation composed of pixels or super-pixels, the regions are iteratively merged in a greedy manner following an application specific heuristic. A particular case of binary partition tree, that we call *binary partition tree by altitude ordering* [24], is obtained when the distance between regions is measured by the minimum dissimilarity between those regions: this is a generalization of single linkage clustering [119, 155, 235] to sparse graphs.

- *Watershed hierarchies* [67, 99, 161, 188]: those hierarchies are constructed by considering the watershed segmentations of an image which is iteratively flooded under the control of an attribute (see Section 3.2 for a more formal definition). For example, the watershed segmentations of the area closings of size k of an image for every positive integer k form the watershed hierarchy by area of the image.

3.1.1 Constructive links between hierarchies.

In [24], we studied the links existing between several important hierarchical image representations: min trees, quasi-flat zones hierarchies, binary partition trees by altitude ordering and watershed hierarchies. This study was done in the framework of edge-weighted graphs and we considered both the relation between those hierarchies computed on the graph or on one of its minimum spanning tree. Moreover, all the results presented in [24] are constructive and can be used to devise practical and efficient algorithms to convert one hierarchical representation into another one when a link was found. The major results are:

- the min tree of the edge-weighted graph and its quasi-flat zone hierarchy are equivalent. This means in particular that the important efforts of the community to develop efficient algorithms to construct min and max trees [64, 85, 126, 153, 157, 172, 185, 193, 224, 260, 262] can be used to compute the quasi-flat zone hierarchy;
- it is equivalent to compute the min tree of the edge-weighted graph and its quasi-flat zone hierarchy on the whole graph or on one of its minimum spanning trees. In other worlds, it is always possible to first reduce a graph to one of its minimum spanning trees and then work on this representation which can lead to further performance improvements and reduce memory usage;
- it is equivalent to compute the binary partition tree by altitude ordering on the whole graph or on one of its minimum spanning trees. The potential benefits are the same as above;
- if the weights of the graph are totally ordered then the binary partition tree by altitude ordering and the quasi-flat zone hierarchy are equivalent. While this condition is usually not fulfilled in practice, it is often useful to assume it at the beginning to facilitate theoretical developments;
- if the weights of the graph are not totally ordered then the quasi-flat zone hierarchy can be obtained by filtering the binary partition tree by altitude

ordering. The quasi-flat zone hierarchy can be seen as a canonical version of the binary partition tree by altitude ordering where all arbitrary ordering choices have been removed;

- any watershed hierarchy induces an ordering of the edge weights such that the binary partition tree for this altitude ordering is equal to the initial watershed hierarchy. This provides a strategy, described below, to compute hierarchical watersheds without computing explicitly each flooding and each watershed segmentation.

3.1.2 Algorithms for some hierarchies of segmentations.

Following the constructive links between hierarchical representations found in [24], we proposed simple and efficient algorithms to compute those representations in [32]. Those algorithms are based on the following observations: 1) a binary partition tree by altitude ordering can be used to construct the quasi-flat zone hierarchy and watershed hierarchies, and 2) a binary partition tree is completely determined by a minimum spanning tree of the graph.

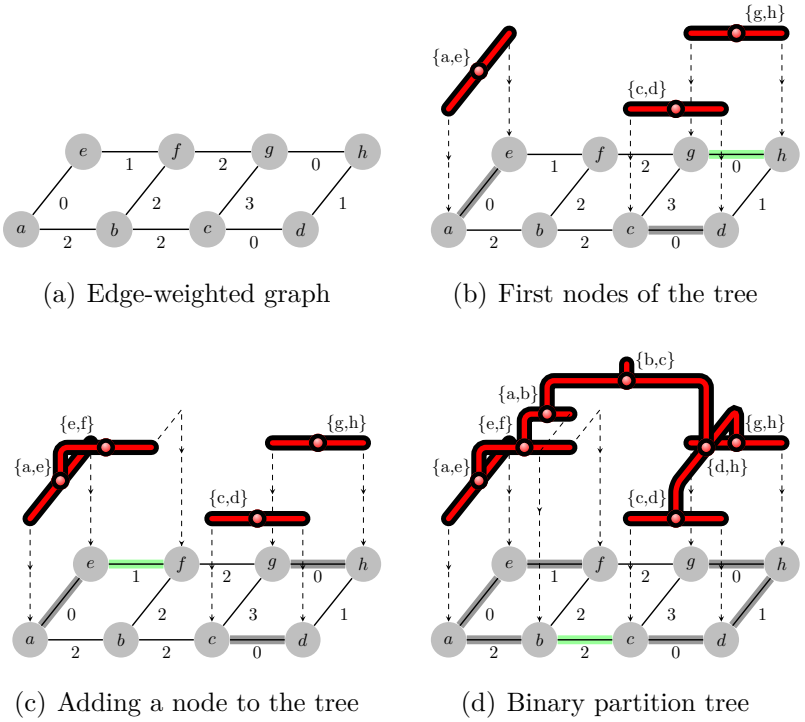
The solution proposed is thus to construct the binary partition tree with a variation of Kruskal's minimum spanning tree algorithm [140] (see Figure 3.3). This algorithm has the same time complexity as Kruskal's algorithm: assuming that the edges of the graph are already sorted by edge weights, it has a quasi linear time complexity $\mathcal{O}(n\alpha(m))$ where n and m are respectively the number of edges and vertices in the graph and where α is the inverse of the Ackermann function [49, 242].

Then, a simple post-processing, with a linear time complexity, is proposed to transform this tree into the quasi-flat zone hierarchy.

The algorithm to compute a watershed hierarchy by a given attribute is more involving. Its very principle is to use the binary partition tree of the graph to find a new ordering of the edge weights of the associated minimum spanning tree such that the binary partition tree of the reweighted graph is equal to the target watershed hierarchy. In other words, we use the binary partition tree to compute the saliency map [100] of the desired watershed hierarchy on the smallest possible graph: the minimum spanning tree of the original graph.

The algorithm proceeds in 5 steps:

1. compute the binary partition tree of the graph;
2. compute the attribute of interest on the tree (for example, a classical attribute is the area of the nodes to obtain a watershed by area);
3. compute the *persistence* values associated to this attribute: the persistence value of a node corresponds to the filtering threshold at which this node ceases to separate two distinct regions in the corresponding filtered image. This can be done with a linear leaves-to-root traversal of the tree;
4. reweight the minimum spanning tree of the graph with the persistence values. This can be done in linear time with a root-to-leaves traversal of the tree; and finally



(a) Edge-weighted graph (b) First nodes of the tree

(c) Adding a node to the tree (d) Binary partition tree

Figure 3.3: The algorithm proposed in [32] to construct the binary partition tree of an edge weighted graph is similar to Kruskal's minimum spanning tree algorithm. It constructs the minimum spanning tree starting by the smallest edges and maintains a collection of trees with a union-find data-structure. Constructing the fusion hierarchy associated to the minimum spanning tree requires to also keep track of the roots of the partial trees when the forest grows.

5. compute the binary partition tree of this reweighted minimum spanning tree.

Assuming that the node attribute can be computed in linear time, the time complexity of this algorithm is dominated by the time complexity of the construction of the binary partition tree.

3.1.3 Simplification of hierarchies of segmentations.

In [9], we studied the problem of removing non-significant regions from a hierarchy of segmentations, while still preserving the hierarchical segmentation structure. Indeed, many algorithms for image segmentation or data clustering contain a step that removes non-significant regions or clusters. One way to achieve such a hierarchical simplification would be to extract all the possible segmentations from the hierarchy, and for each one of them, remove the non-significant regions by merging these regions with one of their neighbors. One issue with this approach is that those merging steps have to be performed in a consistent way at all scales, so that the set of simplified segmentations is still a hierarchy. Another important issue is that such a process would be slow.

To provide an efficient solution to this problem, we proposed to rely on the equivalence between hierarchical segmentations and saliency maps. As shown in Figure 3.4, our algorithm makes use of these different representations to efficiently achieve its goal: it has a linearithmic time complexity $\mathcal{O}(n \log(n))$ with respect to the number of vertices n in the underlying graph. The very idea of the algorithm is that, in the saliency map domain, we can consistently merge two adjacent branches of the hierarchical segmentation by simply setting the weight of a well chosen edge to zero; in other words we force the ultrametric distance to be zero between the target regions.

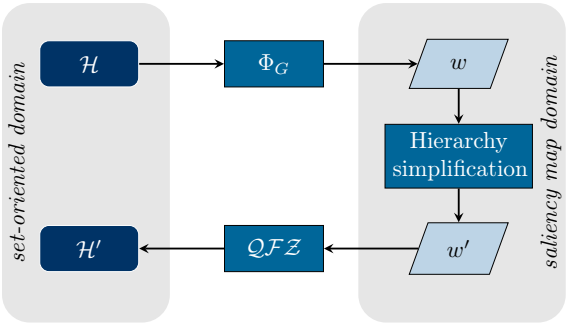


Figure 3.4: A flowchart of the proposed method for removing non-significant regions in a hierarchy of segmentations. A given hierarchy $\mathcal{H}$ is first transformed into its dual representation as a saliency map $w = \Phi_G(\mathcal{H})$. The filtering is then performed in the saliency map domain using information provided by $\mathcal{H}$ (*e.g.*, sizes of the regions). Finally, the filtered hierarchy $\mathcal{H}'$ is defined as the quasi-flat zone hierarchy of the filtered saliency map.

We demonstrated how this method can be applied to improve the quality of hierarchical segmentations on general data and on images with qualitative (see Figures 3.5 and 3.6) and quantitative assessments [9].

3.2 Hierarchy of watersheds

The watershed is a geographical concept [134, 154] studying how flowing waters tends to partition a space along topological ridges. This idea has been transposed to image segmentation [66, 113] where one considers the gradient of an image as a topological relief. Numerous authors have then proposed various definitions of watersheds and their associated algorithms for discrete image analysis [68, 69, 82, 97, 116, 156, 158, 160, 187, 250]. It was also soon noted that watershed segmentations can naturally lead to hierarchies of segmentations [67, 99, 161, 188].

In the following, we will focus on the definition of hierarchical watersheds given in [99] which is based on the notion of watershed cuts defined on edge-weighted graphs [97]. These hierarchical watersheds are of particular interest as they benefit from two fundamental properties: 1) they are the solution of a well defined optimization problem, and 2) they can be computed efficiently in linearithmic time $\mathcal{O}(n \log(n))$ (see Section 3.1).

Indeed, the major difficulty when dealing with optimization problem involving

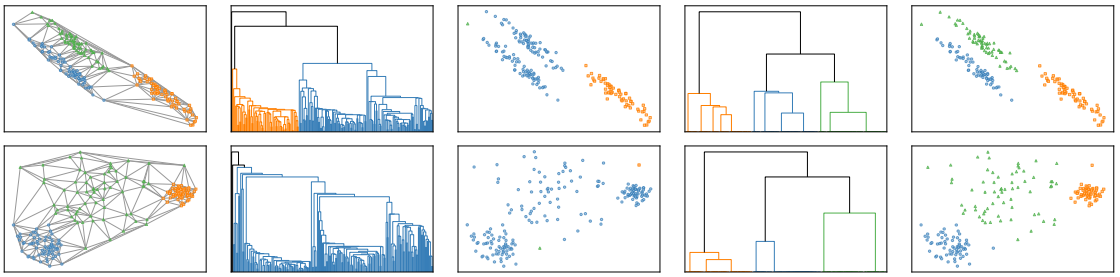


Figure 3.5: Removal of non-significant nodes on the quasi-flat zone (QFZ) hierarchies of two point clouds (first and second lines). For each graph, we show from left to right: the graph with the three ground-truth clusterings of the graph vertices (red, green, and blue), the dendrogram of the quasi-flat zone hierarchy, the clustering into 3 classes for this dendrogram, the dendrogram of the simplification of the quasi-flat zone hierarchy based on cluster size, and the clustering into 3 classes for this simplified dendrogram. Note that the colors used to represent clusterings are arbitrary and do not represent an explicit correspondence between two different clusterings.

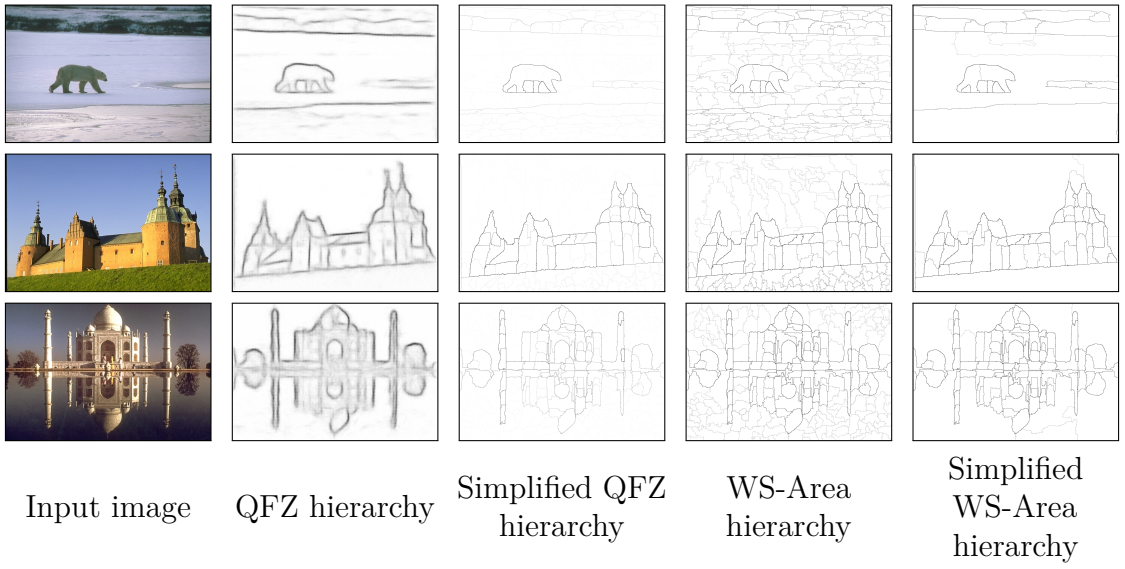


Figure 3.6: Removal of non-significant nodes of the quasi-flat zone (QFZ) hierarchies and of the watershed hierarchies by area (WS-Area) on 4 images of the BSDS 500 dataset [58]. For each image, we show from left to right: the input image, the saliency map of the quasi-flat zone hiearchy, the saliency map of the simplified quasi-flat zone hierarchy based on the region area, the saliency map of the watershed hierarchy by area, and the saliency map of the simplified watershed hierarchy by area based on the frontier strength.

hierarchical segmentations is the combinatorial nature of these representations. Most optimization formulations of hierarchical segmentations lead to NP-hard problems [50, 94, 96, 109, 110, 112, 271] for which classical combinatorial approaches fail to provide scalable solutions. One consequence of this difficulty is that the largest and widely used family of hierarchical segmentation algorithms, agglomerative clustering methods, is heuristic.

Hierarchical watersheds provides a scale-wise optimality, *i.e.*, each flat segmentation corresponding to each scale of the hierarchy is optimal for a specific criterion. Note that, it is not sufficient to have an optimality criterion with an implicit or explicit scale parameter to define a scale-wise optimal hierarchical segmentation: the optimality criterion must also satisfy a scale consistency constraint [128]. In particular, we have shown that classical graph-cut approaches such as min cuts [240], average cuts [251], or shortest path forests [98] do not satisfy this constraint [4]. Ont he contrary, the watershed cut [97] is a non-trivial graph cut method satisfying the scale consistency constraint which leads to the notion of hierarchical watersheds [99].

More formally, given an edge weighted graph $G = (V, E, w)$, we denote by $\mathcal{M} = \{M_i\}$ the set of minima of the map w for the topology of G. A watershed cut of G for a subset $\mathcal{M}'$ of $\mathcal{M}$ is defined as the graph cut associated to the minimum spanning forest of G rooted in the minima in $\mathcal{M}'$ [97, 159]. In other words, we look for the spanning graph of smallest total weight such that each connected component of this spanning graph contains exactly one element of $\mathcal{M}'$.

If we provide $\mathcal{M}$ with a total order relation $\prec$: $M_1 \prec M_2 \prec \ldots \prec M_n$, then, there exists a sequence of cuts $(C_1, C_2, \ldots, C_n)$ such that C_i is a watershed cut for the set of minima $\{M_i, \ldots, M_n\}$ and such that the sequence of segmentations associated to these cuts forms a hierarchy of segmentations (see Figure 3.7). Such hierarchy is called a watershed hierarchy for the ordering $\prec$ [99]. In other words, a hierarchical watershed optimizes a minimum forest problem at each scale of the hierarchy.

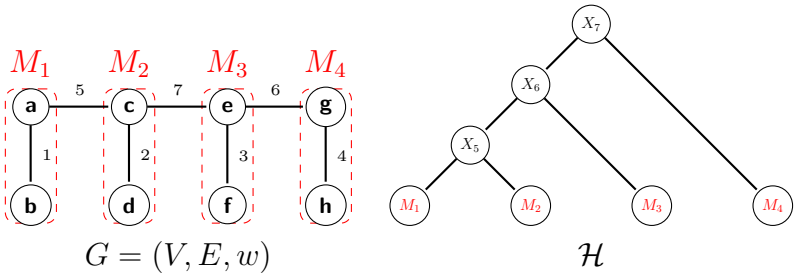


Figure 3.7: A weighted graph $G = (V, E, w)$ with four minima delimited by the dashed rectangles, and the hierarchical watershed $\mathcal{H}$ of G for the minima ordering $M_1 \prec M_2 \prec M_3 \prec M_4$. Note that the graph G is acyclic and that the edge weights are totally ordered: these hypothesis are used to ease the presentation but all the properties and algorithms presented in this section are valid in the general case.

3.2.1 Characterization of hierarchical watersheds

In [5, 29], we studied the following problem: given an edge weighted graph $G = (V, E, w)$ and a hierarchy of segmentations $\mathcal{H}$, determine if $\mathcal{H}$ is a hierarchical

watershed of this graph. In other words, does there exist an ordering $\prec$ of the minima of G such that $\mathcal{H}$ is a hierarchical watershed of G for $\prec$? As observed in Section 3.1, there is a link between the binary partition trees (by altitude ordering) and hierarchical watersheds; in particular, the former can be used to compute the latter. Moreover, we know that, on one side, the binary partition tree is tightly linked to the minimum spanning tree problem and, on the other side, watershed hierarchies are defined as sequences of minimum spanning forests: the main strategy followed to answer to the given problem is then to study how the hierarchy $\mathcal{H}$ relates to the binary partition tree of G and its associated minimum spanning tree.

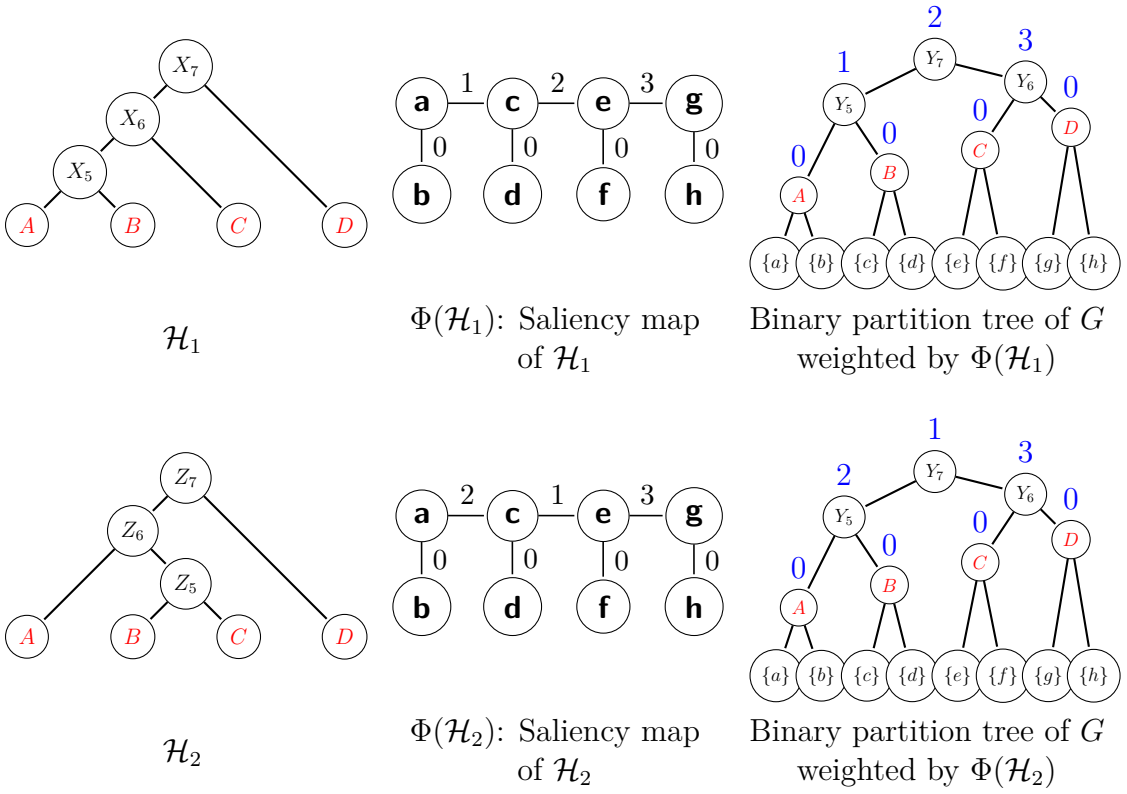


Figure 3.8: $\mathcal{H}_1$ and $\mathcal{H}_2$ are two hierarchies built on the graph G shown in Figure 3.7 and we want to determine if any of them is a hierarchical watershed of G. For both hierarchies, we compute its saliency map (middle column) and we report the saliency values on the binary partition tree by altitude ordering $\mathcal{B}$ of G (third column). We can see that $\mathcal{H}_1$ is one-side increasing for $\mathcal{B}$ and thus $\mathcal{H}_1$ is a hierarchical watershed of G. However, $\mathcal{H}_2$ is not one-side increasing for $\mathcal{B}$: the node Y_7 of $\mathcal{B}$ has the value 1, and the maximal value in its children sub-trees are respectively 2 and 3 which are both greater than 1. Thus, the region Z_5 of $\mathcal{H}_2$ corresponding to this saliency value of 1 cannot exists in any hierarchical watershed of G and $\mathcal{H}_2$ is not a hierarchical watershed of G.

The key element to solve this issue is to study how the values of the saliency map of $\mathcal{H}$ evolve in the binary partition tree $\mathcal{B}$ of G. To do this, we introduced the notion of *one-side increasing map* which states that a map f on the nodes of a binary tree $\mathcal{T}$ is one-side increasing if for any non leaf node n of $\mathcal{T}$, $f(n)$ dominates the values of f on

at least one of the sub-tree rooted in a child of n. Then, the main property of [5, 29] states that a hierarchy $\mathcal{H}$ is a watershed hierarchy of G if the the saliency map of $\mathcal{H}$ is one-side increasing on the binary partition tree $\mathcal{B}$ of G. Figure 3.8 presents a positive and a negative example of this property. Intuitively, if the one-side increasingness of the saliency map of $\mathcal{H}$ on the binary partition tree $\mathcal{B}$ is violated at the node n of $\mathcal{B}$, then, the restriction of the minimum spanning tree of G to the corresponding region of $\mathcal{H}$ cannot be part of a minimum spanning forest at the scale of this region.

Building on this fundamental property, we obtained:

- a constructive characterization of hierarchical watersheds; and
- an algorithm with a linearithmic $\mathcal{O}(n \log(n))$ time complexity for determining if a given hierarchy $\mathcal{H}$ is a hierarchical watershed of the given edge weighted graph G.

3.2.2 Combinatorial analysis of hierarchical watersheds

In [28] we addressed the problem of counting the hierarchical watersheds of a given edge weighted graph G. Indeed, as described above, a hierarchical watershed on G is obtained by choosing a particular ordering of the minima $\mathcal{M}$ of G. It is however possible that several orderings of $\mathcal{M}$ lead to the same hierarchical watershed. For example, the hierarchy $\mathcal{H}$ presented in Figure 3.7 is a hierarchical watershed of the edge weighted graph G for the two following orderings of the minima of G: $M_1 \prec M_2 \prec M_3 \prec M_4$ and $M_2 \prec M_1 \prec M_3 \prec M_4$.

The main contributions of [28] are:

- a definition of the probability of a hierarchical watershed for a given edge weighted graph. It is defined as the probability to obtain this hierarchical watershed if we randomly choose a minima ordering according to a uniform distribution: it is thus equal to the number of mimina orderings leading to this hierarchical watershed divided by the total number of possible minima orderings;
- an algorithm to compute the probability of a given hierarchical watershed for a given edge weighted graph. This algorithm runs in linearithmic $\mathcal{O}(n \log(n))$ time;
- the minimal and maximal probability of any hierarchical watershed on a given edge weighted graph;
- constructive characterizations of the watershed hierarchies of minimal and maximal probability on a given edge weighted graph; and
- an algorithm to find a least/most probable hierarchical watershed on a given edge weighted graph. This algorithm runs in linearithmic $\mathcal{O}(n \log(n))$ time.

3.2.3 Watersheding operator

In [31], we introduced the watersheding operator, which maps any hierarchy $\mathcal{H}$ defined on an edge weighted graph G to a hierarchical watershed $\mathcal{H}_{\mathcal{WS}}$ on G. The

watersheding operator is idempotent and any hierarchical watershed is a fixed point of this operator: the set of fixed points of this operator is thus equal to the set of hierarchical watersheds of the graph. Intuitively, this operator is constructed by inverting the algorithm used for computing hierarchical watershed introduced in Section 3.1. Doing so, we are able to derive a saliency map and an ordering of the minima of G such that the corresponding hierarchical watershed is *similar* to the input hierarchy. We provide an algorithm to compute the watersheding operator in linearithmic $\mathcal{O}(n \log(n))$ complexity. However, the question of obtaining a formal characterization (non algorithmic) of this operator remains an open question.

We demonstrated the interest of the watersheding operator on two applications in image analysis:

- *Refinement of coarse hierarchies.* In [148], the authors propose a high-quality method (COB) to compute hierarchies based on a multiscale gradient learned with deep convolutional methods. While this learned gradient provides very high quality contours for large objects, it tends to ignore small regions which were not part of the ground-truth annotations. In Figure 3.9, we show how the watersheding operator can be used to reintroduced those small regions into such hierarchy.
- *Regularization of hierarchies based on non-increasing attributes.* In a hierarchical watershed, the order in which catchment basins are merged is usually based on an increasing attribute, such as the area or the volume. The constraint of increasingness is very strong and excludes numerous attributes such as geometric descriptors (*e.g.*, compactness or circularity) or more generic significance measures (*e.g.*, energy functionals or learned features). In Figure 3.10, we show how the watersheding operator can be used to find a watershed hierarchy corresponding to a regularized increasing attribute deduced from a non watersehd hierarchy constructed from a non-increasing attribute.

3.2.4 Combinations of hierarchical watersheds

In [4, 5, 27], we studied the combinations of watershed hierarchies from a practical and from a theoretical point of views. In particular, we explored under which conditions the combination of two watershed hierarchies produces a new watershed hierarchy.

Indeed, in [27], we showed that by combining hierarchical watersheds in the combination framework proposed in [100, 101, 137], one can significantly improve the quality of the resulting hierarchy. Figure 3.11 shows an example of hierarchy combination. In this framework, hierarchies are combined thanks to their representations as saliency maps. The combination is done in three steps: 1) compute the saliency maps of the two hierarchies, 2) combine the two saliency maps, and 3) compute the quasi-flat-zone hierarchy of the combined saliency map. The interest of this approach is that combining saliency maps, *i.e.*, functions, is much simpler than combining dendrograms: classical operators such as the point-wise infimum, supremum or average can be used. For example, the results that we presented in [27] showed that be combination by average of hierarchical watersheds by area and by dynamics nearly always improve the

Figure 3.9: First line from left to right: original image I and the gradient G of I computed using the structured edge detector introduced in [114]. Second line from left to right: the saliency map of the COB hierarchy [148] (H_{cob}) of I and three segmentations of H_{cob} with 50, 100 and 200 regions, respectively. We can see that we must go very deep in the COB hierarchy to obtain the internal parts of the person which means that we also get a lot of noise (small non relevant regions). Third line from left to right: the watersheding H_{ws} of H_{cob} and three segmentations of H_{ws} with 50, 100 and 200 regions, respectively. We observe that significant internal regions of the person are closer to the top of the hierarchy.

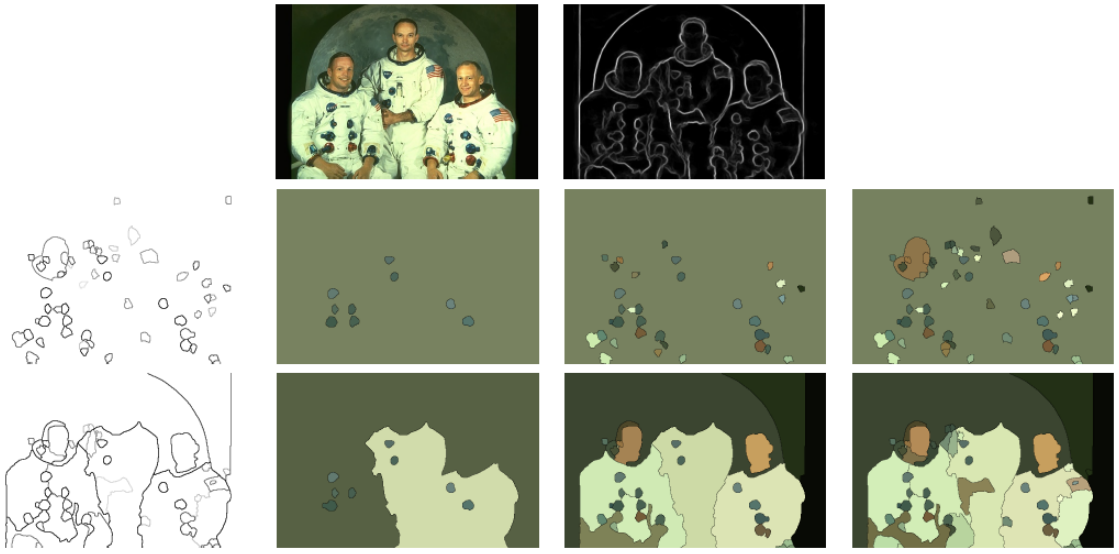


Figure 3.10: First line from left to right: original image I and the gradient G of I computed using the edge detector introduced in [114]. Second line from left to right: the saliency map of the circularity based hierarchy H_{cc}, which is not a hierarchical watershed of G, and three partitions of H_{cc} with 10, 35 and 60 regions, respectively. As expected this hierarchy exhibits mainly circular regions. Third line: the watersheding H_{ws} of H_{cc} and three partitions of H_{ws} with 10, 35 and 60 regions, respectively. This hierarchy combines the information from the gradient and from the circularity based hierarchy: the main regions visible in the gradient and the circular regions are both close to the top of the hierarchy.

performances compared to the individual hierarchies on tasks related to image and object segmentation (see Section 4.1 for a presentation of the evaluation framework we designed for hierarchies of segmentations).

Our contributions in [4, 5] are threefold:

1. we introduced the notion of *flattened hierarchical watersheds* which generalizes the notion of hierarchical watersheds: while hierarchical watershed rely on a total ordering of the minima of the graph, flattened hierarchical watersheds are based on a pre-ordering of the set of minima. We provided a constructive characterization of flattened hierarchical watershed leading to a recognition algorithm running in linearithmic $\mathcal{O}(n \log(n))$ time;
2. we investigated which of the classical combination functions (infimum, supremum, and average) lead to hierarchical watersheds. We showed that the combination by infimum of any two watershed hierarchy is a flattened watershed hierarchy. In other words, this means that the combination of hierarchical watersheds by infimum can indeed be computed directly with a well chosen ordering of the minima, or said differently with a well chosen regional attribute; and
3. we proposed a sufficient condition for a combination of hierarchical watersheds to always result in a flattened hierarchical watershed.

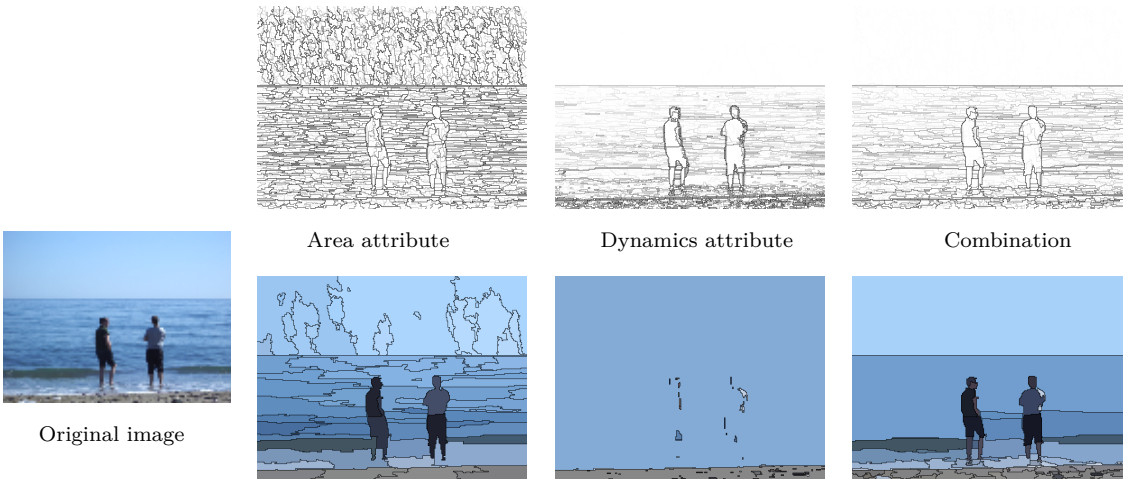


Figure 3.11: Hierarchical watersheds based on area and dynamics and their combination by average. We can observe that, in the watershed hierarchy by area, the main objects of the scene appear close to the top of the hierarchy but large and mostly flat regions such as the sea are over-segmented. On the contrary, in the watershed hierarchy by dynamics the large regions are under-segmented and mainly small regions are close to the top of the hierarchy. By combining the two hierarchies we obtain a better compromise where important objects are close to the top of the hierarchy and the over-segmentation of large regions has been reduced.

3.3 Gradient based hierarchical clustering

Devising algorithms for finding optimal hierarchical segmentation of an edge weighted graph for a given cost function is generally a difficult problem. In the previous Section 3.2, we have seen that hierarchical watersheds are scale wise optimal hierarchies of segmentations where each scale of the hierarchy optimizes a minimum spanning forest problem.

However, most hierarchical segmentation methods used in practice do not optimize an explicit cost function. For example, in the general context of hierarchical clustering, *i.e.*, hierarchical segmentation applied to non image data, the most popular methods probably belong to the family of agglomerative heuristics. They follow a bottom-up approach, in which the vertices of the given edge weighted graph are sequentially merged through some specific strategy. But since the latter is specified procedurally, it is usually hard to understand the objective function being optimized. In this regard, several recent works [89, 94–96, 109, 218, 219] underlined the importance to formulate hierarchical clustering as an optimization problem with a well-defined cost function, so as to better understand how the hierarchies are built.

In this context, Dasgupta [109] introduced a cost function with interesting theoretical properties for evaluating a hierarchical clustering. He showed that optimizing this cost function is NP-Hard and he proposed a heuristic to approximate its optimal solution. The factor of this approximation was later improved by several works, based on a linear programming relaxation [218, 219], a semi-definite programming

relaxation [89], or a recursive ϕ-sparsest cut algorithm [95, 96]. Along similar lines, it was shown that average linkage provides a good approximation of the optimal solution to Dasgupta's cost function [90,174]. A differentiable relaxation inspired by Dasgupta cost function was also proposed [169]. Moreover, a regularization for Dasgupta's cost function was formulated in the context of semi-supervised clustering [93, 249], based on triplet constraints provided by the user.

More generally, the problem of finding the hierarchical clustering whose associated ultrametric/saliency map is close to the given edge weights was extensively studied through linear programming relaxations [50] and integer linear programming [112]. A special case of interest arises when the graph is planar, which is a natural occurrence in image segmentation. By exploiting the planarity of the input graph, a tight linear programming relaxation can be derived from the minimum-weight multi-cut problem [271].

The optimization of hierarchical clustering has thus been the object of numerous works. The main technical barrier in this context comes from the fact that each hierarchical loss function requires a carefully crafted dedicated approximation algorithm. Moreover, despite increasing progress, combinatorial approximation algorithms often struggle to scale to large dataset.

3.3.1 A continuous optimization framework

In [3, 23], we introduced a generic framework to optimize a hierarchical clustering with respect to a given cost function. In the proposed setting, the goal is to find the saliency map that *best* represents the given edge-weighted graph $G = (V, E, w)$. We proposed to express this task as a constrained optimization problem involving an appropriate cost function $J \colon \mathcal{W} \to \mathbb{R}$ defined on the (continuous) space of graph edge weights $\mathcal{W}$, leading to the following formulation

$$\underset{u \in \mathcal{W}}{\text{minimize}} \quad J(u; w) \quad \text{s.t.} \quad u \text{ is a saliency map on } G. \tag{3.1}$$

The saliency map constraint can be formulated as,

$$\forall C \in Cycles(G), \forall e \in C, u(e) \leq \max_{e' \in C \setminus \{e\}} u(e'). \tag{3.2}$$

It is a nonconvex constraint and cannot be efficiently tackled with standard optimization algorithms. We circumvented this issue by replacing the constraint with an operation injected directly into the cost function. The idea is that the saliency map constraint can be enforced implicitly through the operation that computes the subdominant ultrametric, defined as the largest ultrametric below the given dissimilarity function (or equivalently as the saliency map of the quasi-flat zone hierarchy of the graph). One way to compute the subdominant ultrametric is through the min-max operator $\Phi_G \colon \mathcal{W} \to \mathcal{W}$ defined by

$$(\forall \tilde{w} \in \mathcal{W}, \forall e_{xy} \in E) \qquad \Phi_G(\tilde{w})(e_{xy}) = \min_{P \in \mathcal{P}_{xy}} \max_{e' \in P} \tilde{w}(e'), \tag{3.3}$$

where $\mathcal{P}_{xy}$ denotes the set of all paths between the vertices x and y of G. Then, Problem (3.1) can be rewritten as the non-constrained optimization problem

$$\underset{\tilde{w} \in \mathcal{W}}{\text{minimize}} \; J\Big(\Phi_G(\tilde{w});\, w\Big). \tag{3.4}$$

Since the min-max operator is sub-differentiable, it can be seen as a specific max pooling layer, the above problem can be optimized by gradient descent as long as J is sub-differentiable.

3.3.2 Hierarchical cost functions and regularization terms

We proposed several cost functions [3, 23] that can fit in the Problem (3.4). The generality of our framework allowed us to follow the classical approach of combining a data-fidelity term with a regularization term in order to obtain new cost functions. We investigated the following cost functions:

1. *the closest-saliency map fidelity term*, which expresses that the fitted saliency map should be close to the given dissimilarity graph. This is simply obtained by defining the cost function equal to the sum of the squared errors between the sought saliency map and the edge weights of the given graph;
2. *the cluster-size regularization*, which penalizes the presence of small clusters in the upper levels of the associated hierarchical clustering. This term is an adaptation of the very classical area filters found in image processing and relies on the idea presented in Section 3.1 for its generalization to hierarchical clustering. More precisely, this regularization penalizes frontiers of the saliency maps corresponding to the fusion of two unbalanced clusters, *i.e.*, when a very small cluster merges with a very large one, the frontier between the two is pushed toward zero;
3. *the triplet regularization for semi-supervised learning*, which aims to minimize the intra-class distance and maximize the inter-class distance. In this case, we assume that we know some triplets of points such that two points are in the same cluster while the third one is in another cluster. In other words, for each triplet, we know that we should minimize the ultrametric distance between two of the points and maximize the distance to the third one;
4. *the Dasgupta fidelity term*, which is a continuous relaxation of Dasgupta's cost function expressing that the fitted ultrametric should associate large edge weights, *i.e.*, important dissimilarities, to large clusters. To obtain a continuous relaxation of this term, we proposed a continuous relaxation of the size of a cluster in the hierarchy: in other words we proposed a cluster size measure which evolves continuously when the values of the saliency map associated to the hierarchical clustering change.

The use of those cost functions is demonstrated in Figure 3.12.

3.3.3 Efficient algorithms for hierarchical cost optimization

The proposed continuous optimization framework for hierarchical clustering optimization is backed up by specific algorithms [3, 23] used to compute the various terms appearing in the cost functions introduced in the previous paragraph. All the proposed algorithms rely on the properties of binary partition trees by altitude ordering presented in Section 3.1.

The fundamental algorithm is the one that computes the subdominant ultrametric. To obtain an efficient and automatically differentiable algorithm, we observed that the min-max distance between any two vertices x, y is given by the weight of the *pass edge* between x and y, *i.e.*, the edge holding the maximal value on the min-max path from x to y. Moreover, there is a map σ that, for any two vertices x and y, associates the lowest common ancestor of x and y in the binary partition tree of $(G, \tilde{w})$ to the pass edge between x and y (see Figure 3.3.3). The definition of the subdominant ultrametric (Equation (3.3)) can thus be rewritten

$$(\forall \tilde{w} \in \mathcal{W}, \forall e_{xy} \in E) \qquad \Phi_G(\tilde{w})(e_{xy}) = \tilde{w}(\sigma(\mathrm{lca}_{\mathrm{BPT}(\tilde{w})}(x, y))). \tag{3.5}$$

The binary partition tree by altitude ordering can be computed in time $\mathcal{O}(n \log n)$ with a variant of Kruskal's minimum spanning tree algorithm [32] (see Section 3.1). Then, a fast algorithm allows us to compute the lowest common ancestor of two nodes in constant time $\mathcal{O}(1)$, thanks to a linear time $\mathcal{O}(n)$ preprocessing of the tree [62]. The subdominant ultrametric can thus be computed in time $\mathcal{O}(n \log n)$. Note that this algorithm can be interpreted as a special max pooling applied to the input tensor $\tilde{w}$, and can thus be automatically differentiated. Indeed, a sub-gradient of the min-max operator Φ at a given edge e_{xy} is equal to 1 on the pass edge between x and y and 0 elsewhere.

Similar strategies were used do develop differentiable algorithms for the cluster size regularization and the semi-supervised triplet loss running in time $\mathcal{O}(n \log n)$. The computation of the relaxed Dasgupta's cost is more demanding in the worst case; it runs in quadratic time. However when the hierarchy is balanced the computation cost goes down to $\mathcal{O}(n \log n)$: this is a realistic expectation in real applications, especially given the fact that Dasgupta's cost tends to naturally produce well balanced hierarchies.

3.3.4 Validation of the continuous optimization framework

As Problem (3.4) is non-convex, there is no guarantee that a gradient descent method will find the global optimum. To assess the performance of the proposed framework [3, 23], we used the algorithm proposed in [271], denoted by CUCP (Closest Ultrametric via Cutting Plane), as a baseline for the closest saliency map problem. Indeed, CUCP can provide an (almost) exact solution for planar graphs based on a reformulation of the closest ultrametric problem as a set of correlation clustering/multi-cuts [61, 111] problems with additional hierarchical constraints. The results presented in Figure 3.14 show that the proposed approach is able to provide solutions close to the optimal ones (Figure 3.14 a)) using only a fraction of the time needed by the combinatorial algorithm (Figure 3.14 b)), and without any assumption on the input graph.

The computation time of some combinations of cost terms are presented in Figure 3.14 c). *Closest* and *Closest+Size* can handle graphs with millions of edges. *Dasgupta* relaxation is computationally more demanding, which decreases the limit to a few hundred thousands of edges.

3.3.5 Optimal hierarchical clustering on real data

We evaluated the proposed method on real datasets whose size ranges from 270 to 1500 samples [3, 23]. For each dataset, we performed a hierarchical clustering on a k-nearest neighbour graph, and we thresholded the resulting saliency map at the prescribed number of clusters. The analysis is divided in two sets of comparisons: hierarchical clustering (unsupervised), and semi-supervised clustering.

Figure 3.15 a) compares the performance of three hierarchical clustering methods. The baseline is "*Ward*" agglomerative method [252], applied to the pairwise distance matrix of each dataset. The results show that the proposed approach is competitive with Ward method (one of the best agglomerative heuristics). On the datasets Digit1 and Heart, "*Dasgupta*" performs slightly worse than "*Closest+Size*": this is partly due to the fact that our relaxation of the Dasgupta cost function is sensible to data scaling.

Figure 3.15 b) compares the performance of two semi-supervised clustering methods, and an additional unsupervised method. The first baseline is "*Spectral*" clustering [190, 205, 233]. The second baseline is "SVM" classifier [51, 73] trained on the fraction of labeled samples, and tested on the remaining unlabeled samples. The results show that the triplet regularization performs comparably to semi-supervised SVM, which in turn performs better than spectral clustering.

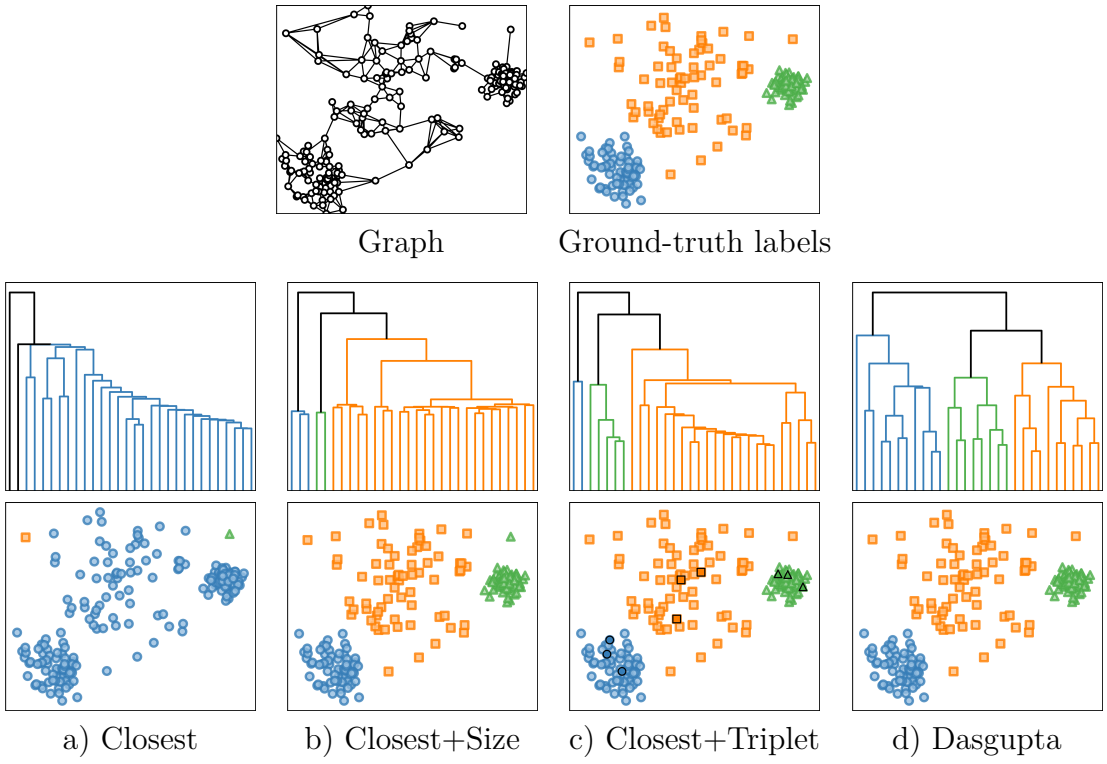


Figure 3.12: Illustrative examples of hierarchical clustering optimization with gradient descent. *Top row*: Input graph and ground truth labeling of the graph vertices. The edge weights are equal to the Euclidean distance between the vertices. *Middle row*: Hierarchical clusterings fitted to the input graph; only the top-30 non-leaf nodes are shown in the dendrograms (all the others are contracted into leaves). *Bottom row*: Assignments obtained by thresholding the hierarchical clustering at three clusters. We can see in a) that fitting the sole *closest* fidelity term leads to an incorrect clustering where very small clusters are present close to the top of the hierarchy. In b), we see that the addition of the *cluster size* regularization term solves this issue. Then, figure c) demonstrates that the semi-supervision brought by known labeled data (vertices marked by squares) is another possible solution. Finally, figure d) shows that the proposed relaxation of the Dasgupta's cost also manages to recover the correct clustering.

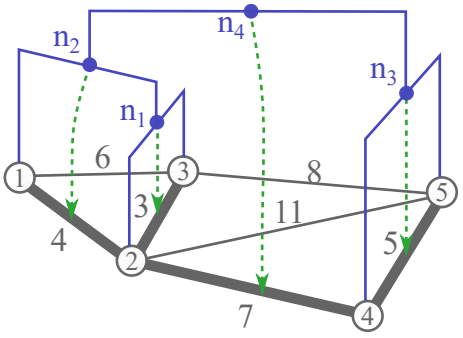
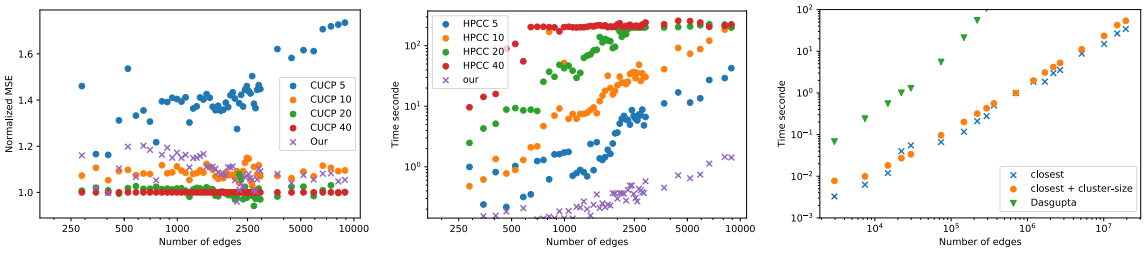
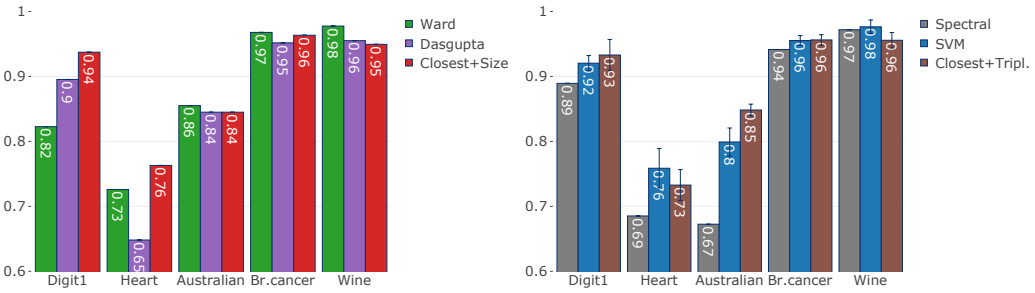


Figure 3.13: Each node of the binary partition tree by altitude ordering (in blue) of the graph (in grey) is canonically associated (green dashed arrows) to an edge of a minimum spanning tree of the graph (thick edges): this edge is the pass edge between the leaves of the two children of the node. The pass edge can be found efficiently in the binary partition tree using the lowest common ancestor operation and the canonical link between the nodes and the minimum spanning tree edges. For example, the lowest common ancestor of the vertices 3 and 5 linked by the edge e_{35} is the node n_4, which is canonically associated to the edge e_{24} ($\sigma(n_4) = e_{24}$): e_{24} is thus the pass edge between the vertices 3 and 5.



(a) Mean square error (b) Computation time (c) Time per iteration

Figure 3.14: Validation and computation time of the proposed continuous optimization framework for hierarchical clustering. Figures (a) and (b): comparison between the CUCP algorithm [271] and the proposed gradient descent approach. For CUCP we tested different numbers of hierarchy levels (5, 10, 20 40). Figure (a) shows the final mean square error (normalized against CUCP 40) with respect to the number of edges in the tested graph. Figure (b) shows the run-time with respect to the number of edges in the tested graph (CUCP was capped at 200 seconds per instance). Figure (c) shows the computation time of the tested cost functions with respect to the number of edges in the graph.



(a) Hierarchical clustering (unsupervised) (b) Semi-supervised clustering

Figure 3.15: Performance of the continuous optimization framework for hierarchical clustering on real datasets.

Chapter 4

Assessment and applications

The contributions presented in the previous chapters are mainly theoretical, methodological, and algorithmic; and they are supported by illustrations, generally on image analysis. While we strongly believe that research must be grounded in strong theoretical foundations, we also think that theory alone is not enough and that the proposed methods must also prove their efficiency in practice. To this end, we have contributed to several application domains:

- *Hierarchy assessment framework:* Hierarchies are abstract and complex representations which are difficult to evaluate. While many works have focused on proposing novel definitions and algorithms to construct hierarchies, only a few works have addressed the issue of assessing those representations. In this context, we have proposed a novel evaluation metric and a comprehensive evaluation framework for hierarchies of segmentations.
- *Applications in bio-medical imaging:* In collaboration with an industrial partner, we have developed several applications related to the analysis of in-vivo imaging of the human skin using reflectance confocal microscopy. This project, which aimed at automatically characterizing skin aging, implied several issues in image analysis and machine learning; some of them were solved using hierarchical approaches.
- *Applications in astronomical imaging:* Astronomers and in particular cosmologists have moved toward a *big data* era since a few decades: the telescopes are producing a very large amount of data every night which need to be analyzed in order to better understand our Universe. In this context, we have worked on the automatic analysis of multiband optical astronomical images for detecting and characterizing astonomical objects.
- *Open source software development:* The development of high quality open source software to disseminate and support research has become an important issue in computer science and neighboring fields. The involvement of the community and private actors into large projects such as *Numpy*, *Scikit-Learn*, *PyTorch* and so on has revolutionized the way researchers work by significantly reducing the time needed to implement and to test complex ideas. In this context, we have developed a novel library which aims at bringing all the

methods related to hierarchical graph analysis to the rich scientific *Python* ecosystem.

4.1 Assessment of hierarchies of segmentations

Hierarchies of segmentations are image representations that are used in a large variety of applications in image processing and analysis: image segmentation [11, 58, 128, 201, 208, 221, 222, 265, 267], occlusion boundary detection [130], image simplification [11, 128, 236], object detection [222], object proposal [201], visual saliency estimation [270].

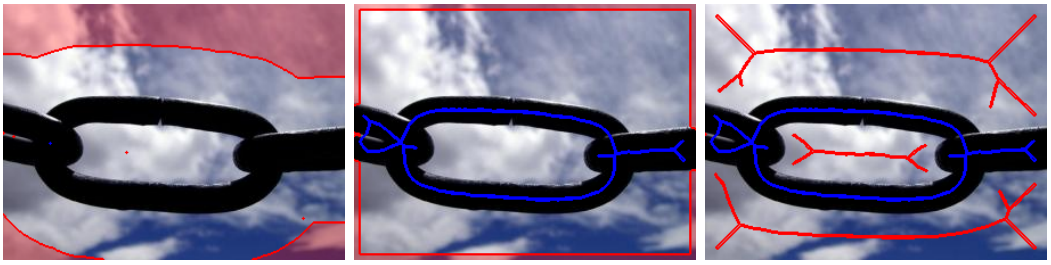
While many hierarchical segmentation methods have been proposed, the question of how to evaluate those representations has only been scarcely studied. A major difficulty is the absence of ground-truth hierarchical segmentations and the difficulty to define such a ground-truth. As a consequence, the evaluation of hierarchical segmentation is currently driven by the specific application of image segmentation where there exist datasets of images with ground-truths such as BSDS 500 [58, 151] or Pascal Context [175]. The authors of [58] have proposed to compare each partition in the sequence of segmentations defining the hierarchy, *i.e.*, the horizontal cuts of the hierarchy, to the ground-truth segmentation. This method leads to classical precision-recall curves. However, the horizontal cuts considered in this framework are just a subset of all possible segmentations that can be constructed from a hierarchy. The authors of [202,203] thus proposed to look for the optimal cut, generally not horizontal, in a hierarchy with respect to a ground-truth segmentation: this assessment provides an upper-bound on the best segmentation that can be extracted from a hierarchy for a given quality measure.

In this context:

- we proposed a novel approach to evaluate hierarchies of segmentation based on object segmentation [35]. Contrarily to existing evaluation methods, our proposition focuses on subsets of regions of the hierarchy which do not necessarily form a cut;
- we proposed a comprehensive evaluation framework for hierarchies of segmentation in the context of natural image analysis [10]. We showed how the different measures collected in this framework help to understand the qualities and defects of hierarchies of segmentations; and
- we utilized our framework to evaluate and optimize hierarchies of watersheds for image and video analysis [10, 33, 35]. This notably allowed us to identify a novel attribute to compute hierarchies of watersheds that outperforms classical attributes.

4.1.1 Hierarchy assessment with object segmentation.

In [35], we introduced a novel assessment strategy for hierarchies of segmentations based on the object segmentation problem for which several datasets with ground-truth labelizations exist: Grabcut [72], Weizmann [52], Pascal Voc [115], and MS-



(a) Erosion-Erosion (b) Skeleton-Frame (c) Skeleton-Skeleton

Figure 4.1: Example of markers generated automatically from a ground-truth segmentation. In each figure, the background and object markers are respectively depicted in red and blue. On the first figure, both markers are obtained by eroding the ground-truth. In the second figure, the foreground marker is a skeleton of the ground-truth while the background marker is only the frame of the image. In the last figure, both markers are skeletons.

COCO [145]. The proposed method evaluates the easiness of finding in the hierarchy a set of regions that represents a semantic object : rather than searching if one can find a segmentation that resembles a human segmentation of the whole scene, we evaluate: 1) if a hierarchy contains a set of regions that matches a given object of the scene, and 2) how difficult it is to find it.

The evaluation relies on a marker based segmentation method proposed in [222] that constructs a two-classes segmentation from a hierarchy of segmentations and two markers: one for the background and one for the object of interest (see online demonstration at `https://perso.esiee.fr/~perretb/ISeg/`). Its principle is to identify the object as the union of the regions of the hierarchy that intersect the object marker but do not touch the background marker. Then, to perform an objective assessment of the different hierarchies, we proposed several automatic strategies to generate object and background markers from the ground truths. Each pair of object/background markers represent a different level of information given to the algorithm (see Figure 4.1). The quality of the binary segmentation obtained from the markers is then evaluated with the F-Measure.

4.1.2 A comprehensive evaluation framework

In [10], we proposed a coherent and comprehensive evaluation framework for hierarchies of segmentations in the context of natural image analysis. It is designed to capture the various aspects of those representations: 1) quality of regions and contours, 2) quality of produced segmentations with horizontal cuts and optimal cuts, and 3) easiness of finding a set of regions representing a semantic object. These measures are evaluated on two types of natural image datasets: 1) Pascal Context as an image segmentation dataset [175], and 2) MS-COCO [145] and Pascal VOC'12 [115] as object segmentation datasets.

The proposed frameworks relies on three categories of measures.

- *Horizontal cuts*: we measure precision-recall curves on both regions and bound-

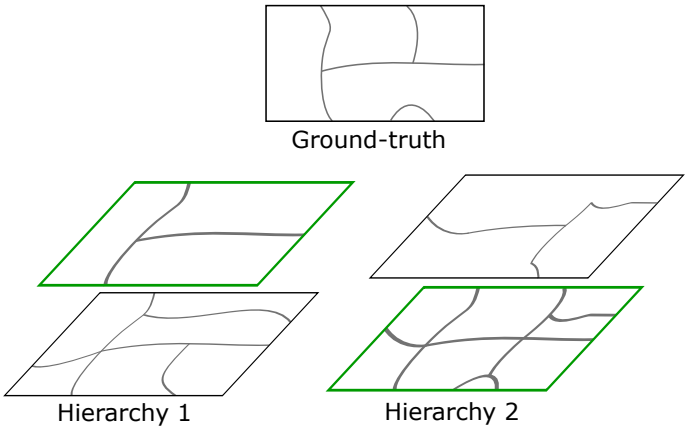
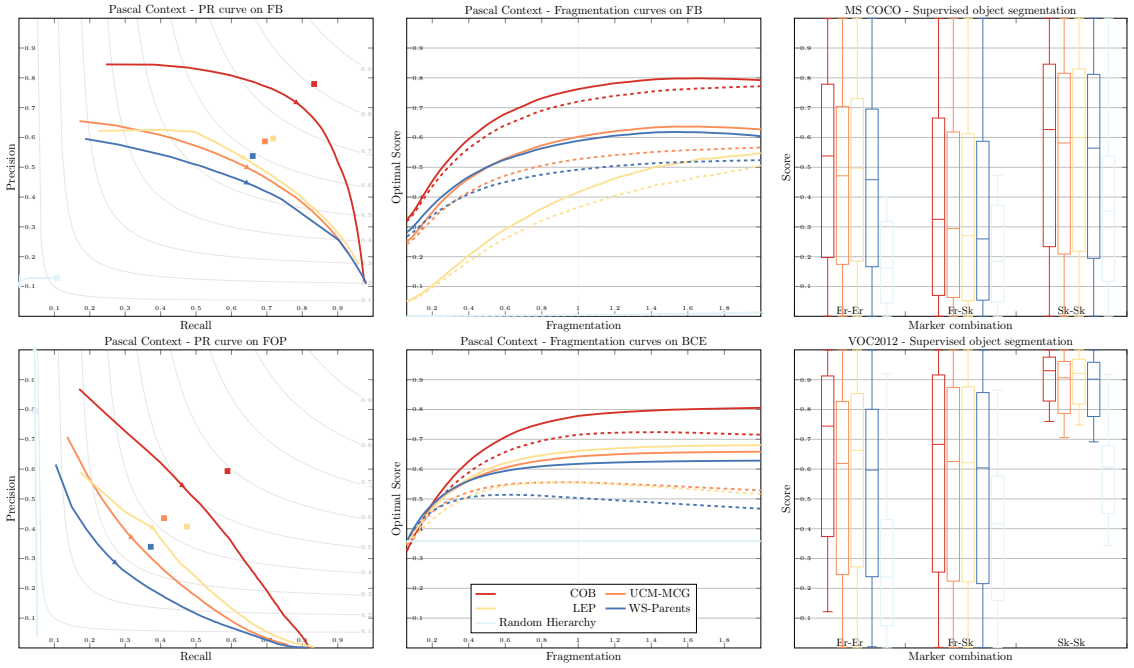


Figure 4.2: Illustration of under- and over-segmentation for hierarchies. Hierarchies 1 and 2 are both composed of 2 levels. Compared to the ground truth, the first hierarchy manages to recover long contours in its coarse level but then fails to recover the other contours at a finer level: the optimal horizontal cut is the coarsest one and the hierarchy is said to under-segment the image. With the second hierarchy the inverse situation happens, the finest partition recovers all the contours of the ground truth but also contains extra-contours. However, the coarsest partition looses the true contours and preserves extra contours: the hierarchy is said to over-segment the image.

aries [58] (as prescribed by [204]). This measure provides global information about the quality of the hierarchy and the consistency of its scales (we say that a hierarchy is scale consistent if the regions of a horizontal cut of the hierarchy represent objects of a same semantic level).

- *Non-horizontal cuts*: we evaluate the quality of the best non-horizontal cuts [202, 203] of the hierarchy with respect to their fragmentation level (ratio between the number of regions in the cut and the number of regions in the ground-truth). This allows us to quantify the upper-bound performances of the hierarchy and provides information about its tendency to under- or over-segment the image (see Figure 4.2).
- *Group of regions*: we evaluate the capacity of the hierarchy to retrieve a given semantic object with respect to different levels of a priori information [35]. This allows to quantify if complex regions of interest can be represented by a small number of regions organized coherently in the hierarchy.

We believe that the proposed framework offers a rich assessment that accounts for the hierarchical nature of the representations and that is not limited to a single use case, which better suits to the wide application spectrum of hierarchies in computer vision and image analysis. The Figure 4.3 shows how the results produced by the evaluation framework can be presented to compare several hierarchical segmentation methods.



	ODS		FOC		ODM		Mean score
	FB	FOP	FB	BCE	COCO	VOC	
COB	0.749	0.499	0.705	0.722	0.681	0.919	0.713
UCM-MCG	0.562	0.342	0.556	0.610	0.615	0.864	0.592
LEP	0.579	0.390	0.378	0.626	0.632	0.881	0.581
WS-Parents	0.528	0.279	0.549	0.592	0.594	0.847	0.565
Random	0.117	0.085	0.005	0.358	0.312	0.535	0.235

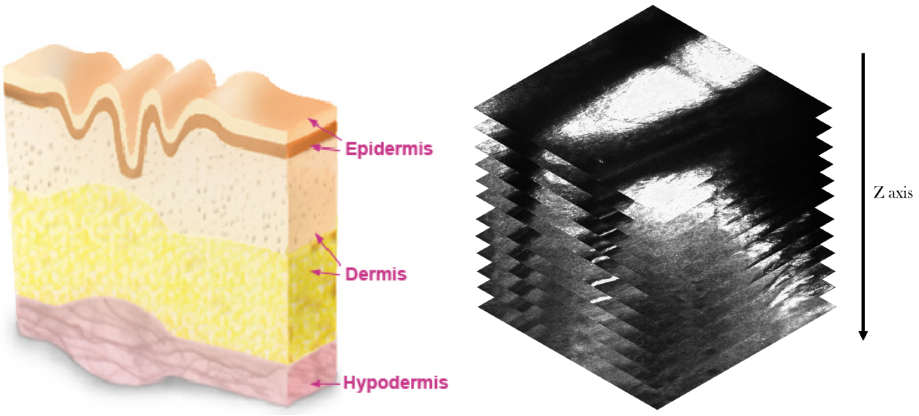
Figure 4.3: Example of comparison between the watershed hierarchy by number of parent nodes and other state-of-the-art methods (a random hierarchy is used as a baseline). **Precision-recall (PR) curves** for boundaries (FB) and regions (FOP) based measures on Pascal Context (first column): each curve represents the variation of precision and recall for the different partitions of the hierarchy. **Fragmentation–Optimal Cut score curves (FOC)** for boundaries (FB) and regions (BCE) based measures on Pascal Context (second column): each plain curve represent the upper-bound score achievable for a given fragmentation value. The corresponding dashed curves represent the score obtained by horizontal cuts. **Supervised object detection** on MS-COCO and Pascal VOC'12 (last column): for each method and each combination of markers, we see: 1) the median F-measure (central bar), 2) the first and third quartile (extremities of the box), and 3) the lowest datum still within 1.5 inter quartile range (difference between the third and first quartile) of the lower quartile, and the highest datum still within 1.5 inter quartile range of the upper quartile range (lower and upper extremities). The principal performance measures are summarized in the table: F-Measure of FB and FOP scores at the Optimal Dataset Scale ODS (precision-recall curves), Area Under the Curve of FOC for FB and BCE (fragmentation curves), and Object Detection Median score (ODM) on MS-COCO and Pascal VOC'12 datasets (supervised object detection). The last column of the table gives the mean score of each method.

4.1.3 Evaluation of hierarchical watersheds

In [10, 33, 35] we have utilized our evaluation framework to assess and optimize the performances of morphological hierarchies of segmentations, namely the quasi-flat zone hierarchy and watershed hierarchies. The main results of this study are the followings.

1. *Importance of the gradient measure.* Those hierarchies are constructed on edge-weighted graphs, where edge-weights represent dissimilarity values between pixels. The most simple gradient measures use only colorimetric information: in this category, we tested an Euclidean distance in the RGB color space and an Euclidean distance in the Lab color space, which is supposed to be more compliant with human color perception [125]. Beyond simple colorimetric gradient, recent advances in machine learning have led to supervised contour detectors: in this category, we consider the Structured Edge Detector (SED) from [114]. We do not observe a clear improvement with Lab gradient compared to RBG gradient. However, there is almost always a large gain by switching from a local RGB or Lab gradient to the supervised semi-local gradient SED.
2. *Importance of the post-processing.* In Section 3.1, we present a hierarchy filtering method [9] that can be used to post-process the candidate hierarchies. We observe that such filter is crucial for quasi-flat zone hierarchies and watersheds by dynamics which generally suffer from having a lot of very small regions close to the top of the hierarchy.
3. *Best attribute for hierarchical watersheds.* Hierarchies of watersheds are constructed by filtering the gradient according to an attribute such as the dynamics, the area, or the volume. In this study, we identified a novel attribute, *the number of parent nodes*, that consistently outperforms traditional attributes. According to the mean score, the best tested attributes are (from best to worse): number of parent nodes, volume, area, and dynamics.
4. *Comparison to state-of-art methods.* The properties of the best morphological hierarchies are discussed and compared to state-of-the-art approaches proposed in the computer vision field. As reference methods, we include Multiscale Combinatorial Grouping hierarchies [201] (MCG), Convolutional Oriented Boundaries hierarchies [148] (COB), and Least Effort Segmentation [274] (LEP) in our assessment. The best method, by a large margin, is COB which is the only one that uses a deep learning based contour detector. Hierarchical watersheds are then slightly worse in terms of mean score compared to MCG and LEP but are competitive for measures based on non-horizontal cuts (see Figure 4.3).
5. *Computation time.* Regarding execution times, the watershed approach is at least an order of magnitude faster than other methods with a mean execution time of 90 ms on images of size 481×321 pixels for the hierarchical watershed methods against 800 ms for COB (with a GPU), 2 s for LEP, and 24 s for MCG.

In conclusion, we see that, used in conjunction with a state-of-the art contour detector, watershed hierarchies are competitive with complex state-of-the-art methods for



(a) Skin layers. (b) RCM stack.

Figure 4.4: a) Schematic view of the skin layers organization. b) A stack of reflectance confocale microscopy images provides a non-invasive observation of the skin up to the dermis with a lateral resolution sufficient to see individual cells.

hierarchy construction based on the same gradient information, but they are much faster to compute.

4.2 Automatic characterization of skin aging in reflectance confocal microscopy

The characterization of skin conditions is a major challenge for skin aging understanding and cosmetic research. Skin aging is defined by a set of alterations of its various components over the years. Those alterations are governed by intrinsic and extrinsic parameters such as genetic and environmental factors. The two major layers of the skin, the epidermis and the dermis, are both significantly affected by skin aging (see Figure 4.4 a)).

In a collaboration with Clarins Laboratories we investigated how skin aging can be automatically characterized with reflectance confocal microscopy. Indeed, reflectance confocal microscopy opens a window into the skin to see the epidermis and the dermis non-invasively up to a depth of 200 μm. The representation of the cells in a layer with a thickness less than 5.0 μm can be achieved with keratin, melanin and dermal fibers working as natural contrast agents [263]. The images obtained non-invasively have a resolution of 0.5 to 1.0 μm in the lateral axis, *i.e.*, close to that of histology [197, 248] (see Figure 4.4 b)). Several descriptors related to skin aging have been highlighted from confocal images [146, 264]: thickness of the epidermis, epidermal state characterized by a regular or irregular honeycomb pattern, shape of the dermal-epidermal junction, and pigmentation and alteration of the collagen fibers in the dermis. These descriptors have been correlated with histological aging [248] and a semi-quantitative score of aging has been established, requiring visual assessment of the images by experienced dermatologists [146].

We studied how this visual assessment can be automatized in the following works:

1. epidermal state characterization [43] with a hierarchical segmentation of the cells and the supervised classification of the cells as regular or irregular;
2. dermal-epidermal junction detection [16, 43] with 3D conditional random fields incorporating a biological a priori;
3. dermal-epidermal junction characterization [44] with hierarchical pattern spectra; and
4. collagen fiber characterization [48] with supervised texture classification.

Those developments were validated in a clinical study involving 160 subjects where we demonstrated that the features automatically extracted by the proposed methods show good correlation with the expert evaluator. To our knowledge, these are the first results comparing a computer-based approach to dermatologists' annotations for the assessment of skin aging using in vivo confocal microscopy.

The two following sections give a more detailed presentation of two of those works which rely on hierarchical image analysis: the epidermal state characterization and the dermal-epidermal junction characterization.

4.2.1 Epidermal cells segmentation for skin aging

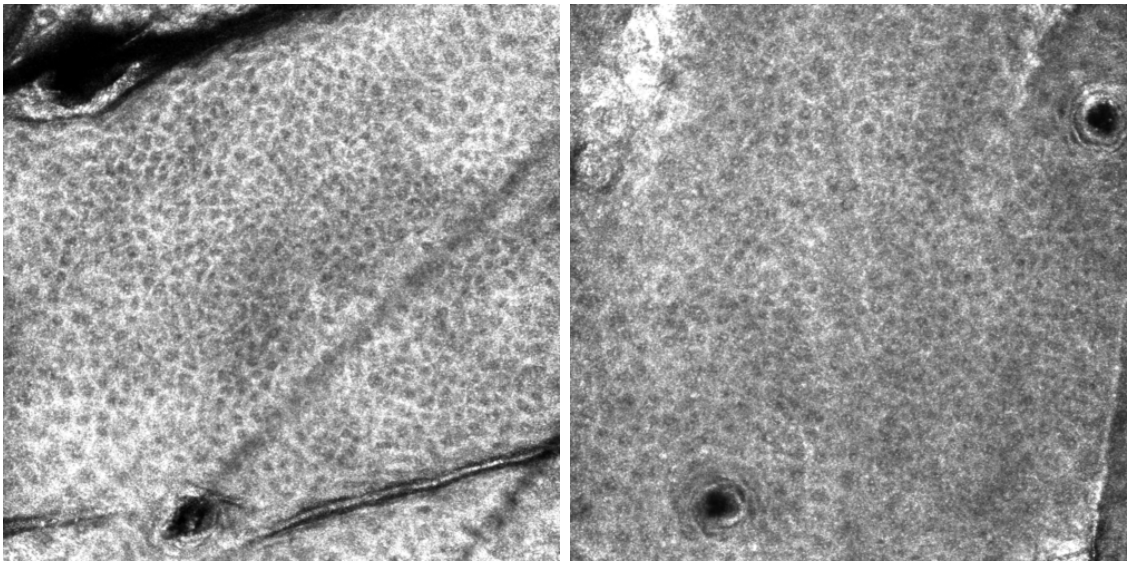
The epidermis is the outer layer of the skin which serves as a physical and chemical barrier against the environment. Cells start from the lower layer of the epidermis and migrate upward to the outer layers. The cells mature throughout this process: they become larger and ultimately end up as a compact anucleated natural moisture barrier. The cell desquamation on the skin surface is compensated by the renewal of the epidermis, a process undertaken by the keratinocytes. The epidermal keratinocytes appear as outlined cells, which form a honeycomb-like pattern (see Figure 4.5). With skin aging, the keratinocytes show increased variability in size and shape.

We utilized a hierarchical watershed by area to obtain a hierarchical segmentation of the image. The final cell segmentation (see Figure 4.6) was extracted from the hierarchy using a priori knowledge on the minimal and maximal cell sizes, which was correlated with histological data. After the segmentation of the keratinocytes, each cell is classified by a random forest classifier as regular or irregular based on its shape and neighborhood. Finally, we proposed two global image descriptors based on the amount and size of cells classified as regulars.

Our results showed significant differences between young and old populations, most significantly on the volar arm. We also observed that the honeycomb pattern irregularity is increased by 22% in sun-exposed area among young subjects, which indicates that irregular honeycomb pattern due to sun exposition can potentially be quantified using our method.

4.2.2 Dermal-epidermal junction characterization

The dermal-epidermal junction is a 2D surface separating the epidermis from the dermis which undergoes multiple changes under pathological or aging conditions. Its peaks and troughs, called dermal papillae, are due to projections of the dermis into

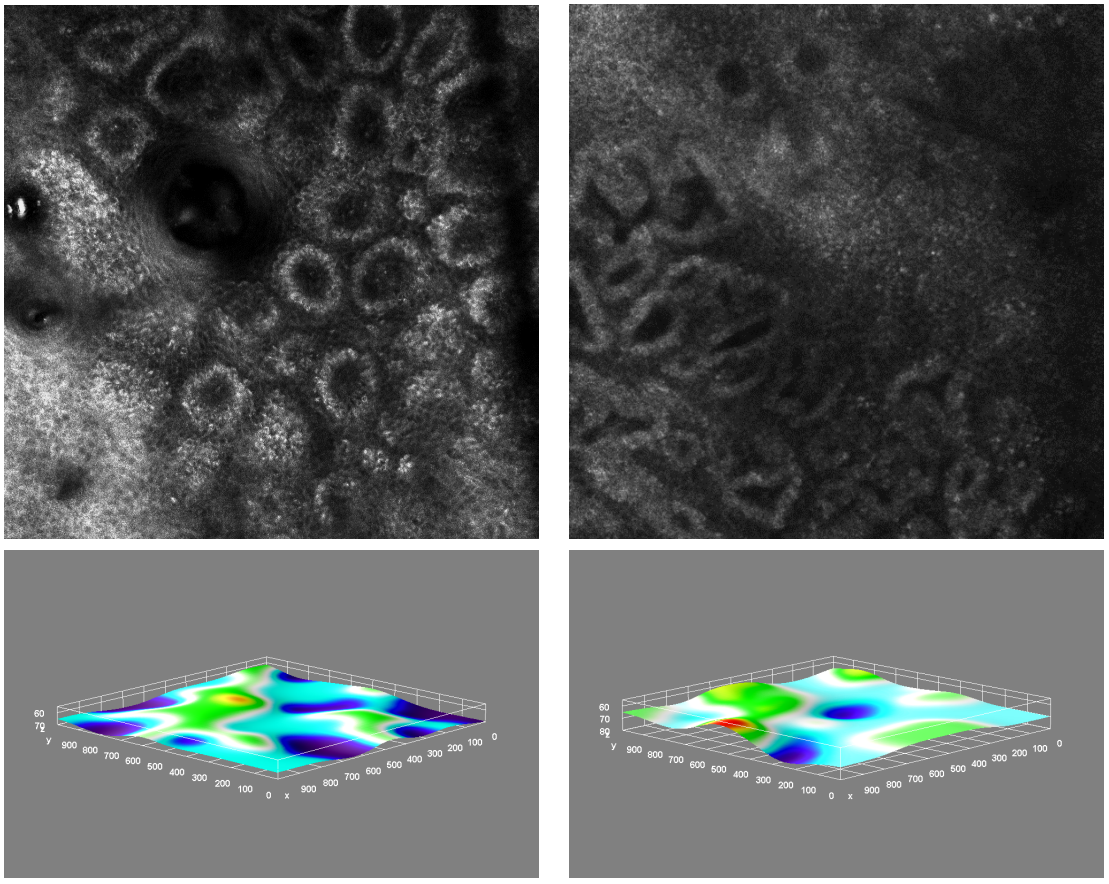


(a) Young epidermis (b) Aged epidermis

Figure 4.5: Epidermal honeycomb pattern of a young (a) and an older subject (b). One can notice that the images are affected by several defects: noise, blur, non-homogeneous intensity, and artifacts like pores and hairs.

(a) Epidermis image (b) Cell segmentation (c) Regular cells

Figure 4.6: Illustration of the epidermal cell classification into regular or irregular class. A segmentation of the individual cells is first computed. Each cell is then classified based on geometric features and its spatial neighborhood.



(a) Young subject (b) Aged subject

Figure 4.7: Dermal-epidermal junction aging. First row: en-face sections of 2 relectance confocal microscopy stacks coming respectively from a young (a) and an aged (b) subject. The dermal-epidermal junction corresponds to the brighter rings. Second row: automatically segmented dermal-epidermal junction surfaces from a young (a) and an aged (b) subject using our method [16, 43].

the epidermis. With skin aging, the appearance of this junction flattens, which has important consequences such as lower epidermal adhesion (see Figure 4.7).

In [16, 43], we proposed a method to obtain reliable segmentation of the dermal-epidermal junction in reflectance confocal microscopy images. Moreover, this method guarantees that the segmented region of the 3D stack of images is indeed a 2D topological surface defined on a regular grid, *i.e.*, an elevation/topographic map: it can thus be characterized with classical morphological tools such as granulometries and pattern spectra [149]. The principle of these tools is to iteratively apply a sequence of increasing filters and to measure the evolution of the filtered image. They have been extended to attribute connected filters [105, 246] (see Section 2) to obtain efficient and powerful multi-scale features using tree based representations of the images.

In [44], we provided a method for automatically characterizing a dermal-epidermal junction surface in order to estimate the aging process. The proposed method relies on the extension [106, 147] of pattern spectra to the self-dual image representation

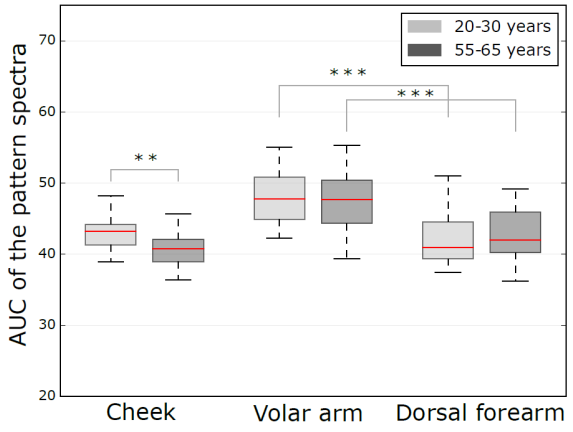


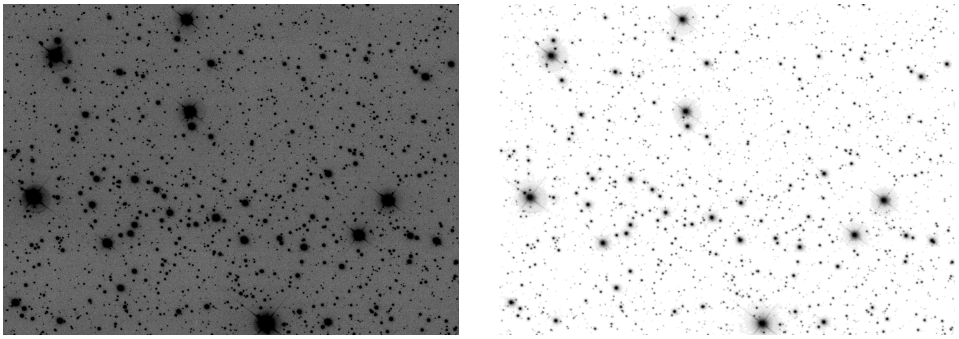
Figure 4.8: Area under the curve of the pattern spectra of the dermal-epidermal junctions. The box-plot shows the distributions of the area under the curve of the pattern spectra for each location and age group. We see that there is a significant (**) difference between the two age groups on the cheek (chronological aging). There is also a significant difference (***) between the volar arm (photo-protected) and the dorsal forearm (photo-exposed) in both aged groups showing the effect of photo-aging. (Statistics significance are defined as follow: **: $0.001 <$ P-values ≤ 0.01, ***: P-values ≤ 0.001.)

called the tree of shapes [167]. In order to adapt the method to our problem, we generalized the subtractive filtering rule [246] to handle non increasing attributes with the tree of shapes and we defined a novel measure in order to characterize the filtered surfaces. The method was assessed on a specifically constituted dataset and we showed that the proposed surface feature significantly correlates with both chronological aging and photo-aging (see Figure 4.8).

4.3 Astronomical image analysis

Since a few decades, astronomy and in particular cosmology, relies on the production of large automated survey of the skies. The idea behind this approach is that the observation of a very large number of objects is necessary to obtain reliable statics, especially on galactic populations and their characteristics. Those statistics are then compared to the predictions made by theoretical cosmological models in order to validate, discard or refine them. The analysis of such surveys is thus an important step toward the understanding of how the Universe evolves and the physics governing this evolution. One of the key feature of an astronomical survey is its *depth*, *i.e.*, its ability to detect objects with a low surface brightness which usually correspond to far objects. Indeed, as the speed of light in the void is fixed and finite, the further an object is, the older it is: observing those far objects thus offers us a direct view of the past of the Universe.

For example, in the Sloan Digital Sky Survey [272], a dedicated 2.5 m wide-angle optical telescope was used during 10 years to scan 35% of the full sky in five optical bands (spectroscopic acquisitions were also made): it covers about 1 billion objects.



(a) Astronomical image (b) Astronomical objects

Figure 4.9: a) Astronomical image from the Sloan Digital Sky Survey (size: 2048×1489 pixels, 32 bits per pixels): the contrast has been inverted and adjusted manually to emphasize faint sources. b) Multilevel segmentation map.

The depth of the Sloan Digital Sky Survey allows us to see the furthest objects as they were 7 billion years ago, *i.e.*, roughly at half the age of the Universe.

One important step in astronomical image analysis is source detection where the goal is to detect all the bright astronomical objects that shine on a dark background (see Figure 4.9(a)). Despite the apparent simplicity of the problem, the difficulty here relies mainly on the large variations in scale (from few pixels to thousands of pixels) and in brightness (from less than 0 dB to dozen of dB) of the various sources. This leads mainly to two difficult cases: the extraction of very faint sources drown in the background noise and the separation of overlapping sources. One can also note the presence of artifacts that should not be detected as sources, like the diffraction spikes around the bright stars that are caused by the support of the secondary mirror of the telescope.

In this context, we contributed to the following works which are described in the next sections:

- faint astronomical sources detection [8, 17, 25] with multi-scale Markovian models;
- multiband astronomical image segmentation [2, 46] with component-graphs (generalization of max trees to multiband images); and
- object detection in multiband galaxy images with max trees [14, 21, 37].

4.3.1 Astronomical source detection with hierarchical Markovian models

In [8,17,25] we proposed to use hierarchical Markovian models to segment faint sources in multiband astronomical images. Hierarchical Markovian models were introduced in [74]. Similarly to hidden Markov chains, and contrarily to traditional hidden Markov random fields [122, 256] used in image analysis, in hierarchical Markovian models, it is possible to perform an exact inference based on the maximum of Posterior Marginal criterion with a 2-pass algorithm [74, 227]. Those hierarchical models have

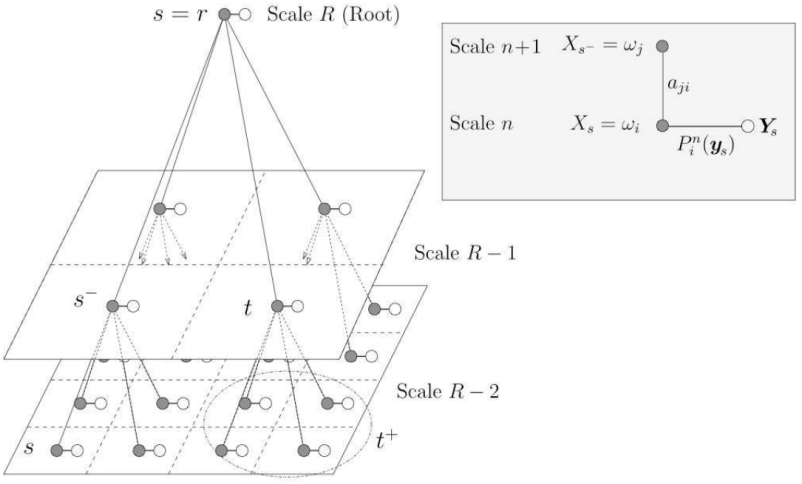


Figure 4.10: Dependency graph of a hierarchical Markovian model corresponding to a quad tree structure. Filled black circles represent labels (X) and white circles represent observations ($\mathbf{Y}$). At the bottom of the quad-tree the observation at each leaf node of the tree is the mutivariate pixel value at the corresponding position. Whereas the information contained in the observations is propagated upward within the quad-tree, the labels are propagated downward according to Markovian transmission probabilities (a_{ij}), *i.e.*, a label at a given level of the quad-tree only depends on its parent.

been applied to quad trees [141, 207] (see Figure 4.10) or similar structures [143]. Both approaches, on chains [199, 200] and quad-trees [170], have also been refined through pairewise or triplet models.

In [17, 25], we proposed to use a hierarchical Markovian model on a quad-tree to segment sources in astronomical images. In order to obtain a completely unsupervised segmentation method, we modeled the data likelihoods (the distribution of pixel values inside a given class) with multivariate Gaussians and we relied on an Expectation–Maximization algorithm to iteratively estimate the model parameters (data likelihoods and class transition probabilities) and the corresponding maximum of posterior marginal.

To evaluate the performances of the proposed method we applied it to simulated data and we showed that we were able to detect galaxies down to a signal-to-noise ratio of 1.5 (peak signal to noise ratio). The method was also applied and validated on a real image dataset and compared to the reference method SExtractor [65] showing that we were able to extract more faint objects while keeping a lower false detection rate.

In [8] we extended the hierarchical Markovian model to general trees and applied it to max tree image representations. Replacing the quad tree by a max tree offers several benefits:

- morphological trees are naturally adapted to each image, removing the presence of the block artifacts of the quad tree [207];

- they offer the possibility to integrate various node attributes as multivariate observations. More specifically, all the shape descriptors that have no meaning in a quad tree are now usable; and
- observations are naturally present at every level of the tree, whereas we usually only have data at the leaf level in a quad tree in practice.

Figure 4.11 illustrates how the use of the max tree improves the result compared to the quad tree in astronomical source detection. One can note that this model was also applied to the problem of blood vessel segmentation in eye fundus images where we managed to obtain state-of-art results (best method on one dataset, and second best on the other) without using any supervision [8].

4.3.2 Astronomical source detection with mutivariate component graphs

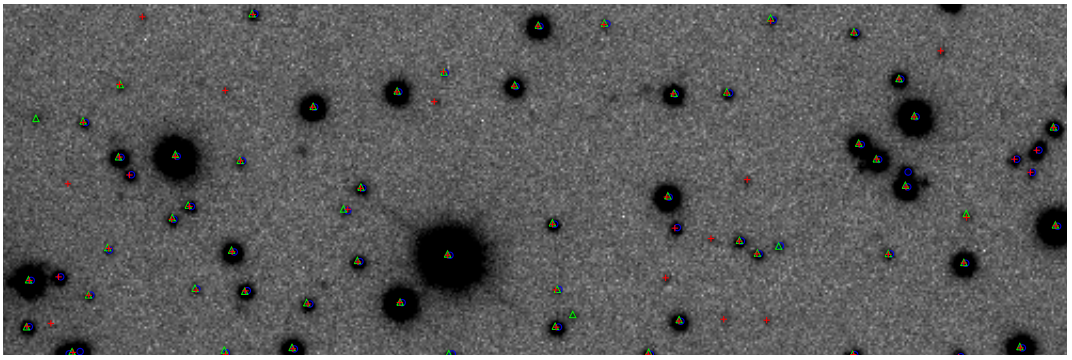
In [2, 46], we developed a new method to perform source detection in multiband astronomical images using component graphs, a natural extension of the max tree to multiband images.

In the framework of mathematical morphology, component trees and component graphs are classical structures for image modeling and analysis. The component trees (min trees, max trees [133, 224], trees of shapes [167]) benefit from fast, efficient construction and varied filtering algorithms. However, they are limited to single-band image processing. Extension to multi-band image processing usually requires a total vectorial order that is application-dependent [37, 178]. On the other hand, the component graph [179, 180] is designed to handle multiband images by relying only on a partial ordering of the pixel values at the cost of a higher complexity [196]: the component graph is not longer a tree but a directed acyclic graph.

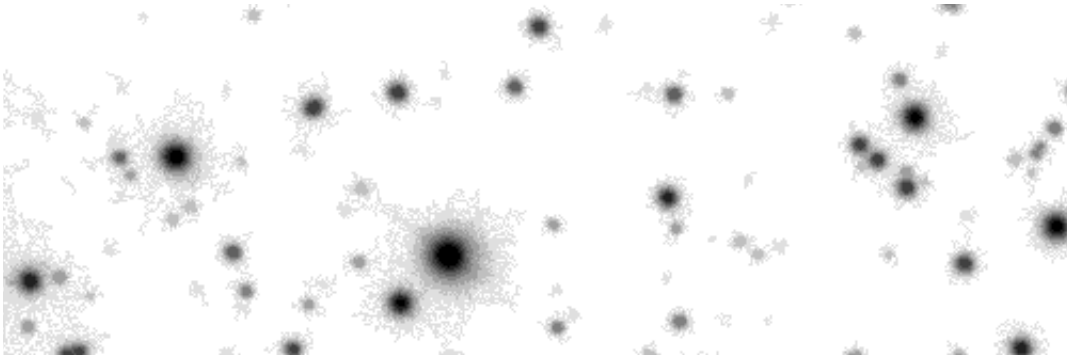
In [45, 243], the authors proposed a novel source detection method for single band astronomical images relying on statistical testing: significant components of the max tree are separated from noise thanks to hypothesis testing and the significant components are then clustered into objects by attribute connected filtering. The motivation to extend this method to the component graph is twofold: 1) the multiband information will naturally increase the amount of information available for statistical testing (for a given object size, fainter objects can be detected), and 2) the multiband information can help to deblend overlapping objects.

While the transposition of the statistical testing method to the component-graph is mostly straightforward, the clustering of significant nodes in the component-graph is much more challenging (see Figure 4.12). To do so, we proposed two complementary filtering strategies, each addressing a particular situation:

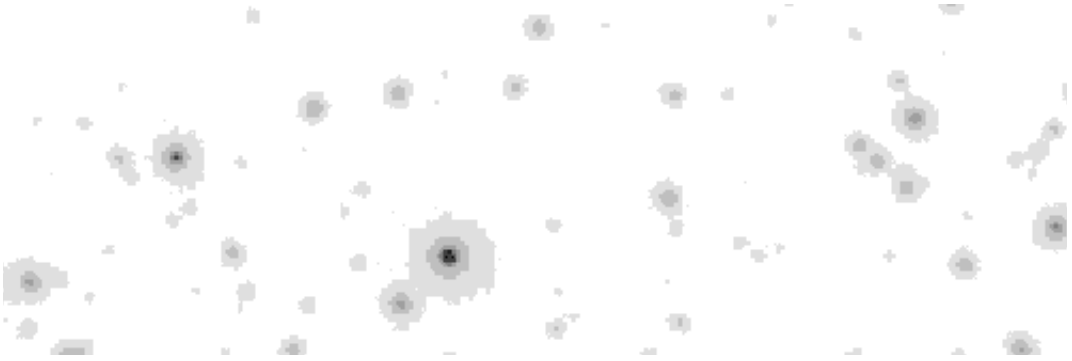
- *duplicated object detection*: in a component graph, a same object can be represented by several nodes, those nodes can be organized hierarchically (as in a max tree) but might also be incomparable (they are then in different branches of the component-graph). To detect such configurations we relied on the fact that astronomical objects usually have a well identified center and we considered that components sharing the same center indeed represent the same



(a) Detected sources

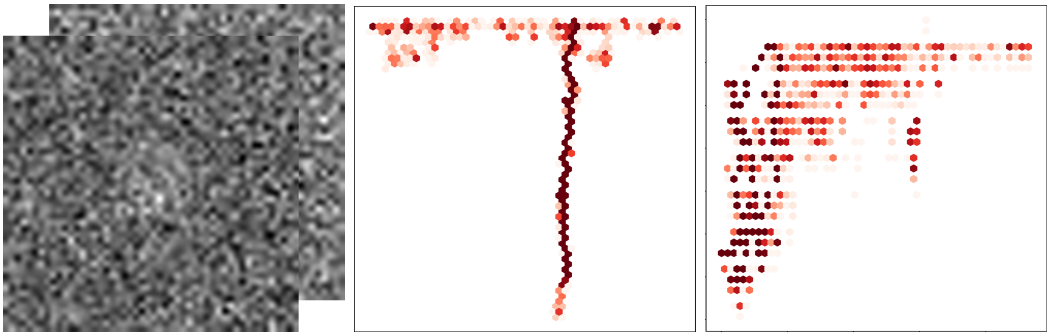


(b) Markovian segmentation on a max-tree



(c) Markovian segmentation on a quad-tree

Figure 4.11: Comparison of the proposed astronomical source detection algorithms with the reference method. (a) Close up view of Figure 4.9(a). Red crosses: sources detected by a hierarchical Markovian model on a max tree, green triangles: sources detected by a hierarchical Markovian model on a quad tree, and blue circles: sources detected by the reference method SExtractor [65]. (b) and (c): Segmentations map obtained respectively with a max tree quad tree Markovian classification. We can observe that the max tree approach better fits the faint details of the image: it is able to detect more sources.



(a) 2-band image (b) Max-tree (c) Component-graph

Figure 4.12: Comparison between the max tree and the component graph of a simulated astronomical image containing a single faint source. The max tree is built on the first band of the simulation while the component-graph accounts for its two bands. The color map encodes the level of significance of the nodes: the darker a node is, the more its content differs from noise. In the max tree, we can easily identify a long branch of very significant nodes corresponding to the single source present in the image. In the component-graph, the situation is much more complicated with a thick *column* of nodes corresponding to the astronomical source.

object;

- *partial object detection*: in a component graph, the overlapping of two different objects can appear as a significant component separated from the two initial objects. To detect this configuration, each significant component is eliminated if it has a spatial neighbor already identified as an object.

The proposed method was validated on a simulated 3-band astronomical image datasets and showed superior performances in terms of precision-recall compared to the single band approach (see Figure 4.13).

4.3.3 Object detection in multiband galaxy images

In [14, 21, 37], we developed a method to detect specific structures in multiband astronomical images of galaxies using the max tree. This method was part of a larger project about automatic galaxy image characterization. Indeed, galaxies are complex objects and their shapes are mainly the results of their various gravitational interactions (collision and fusion between galaxies). Galaxies are usually classified according to the Hubble scheme which comprises two main classes: elliptic galaxies and spiral galaxies. Elliptic galaxies are essentially structureless spheroids but spiral galaxies are much more complex; their internal structure comprises several components such as an inner bulge, a flat outer disc, a varying number of spiral arms, and so on. A possible way to characterize these spiral galaxies is through their decomposition into their internal structures. To do so, direct parametric models of these components were proposed and the decomposition of the galaxy is obtained by solving the inverse problem of fitting the parametric model to the observation. However, the complexity of parametric models is limited and they often struggle to

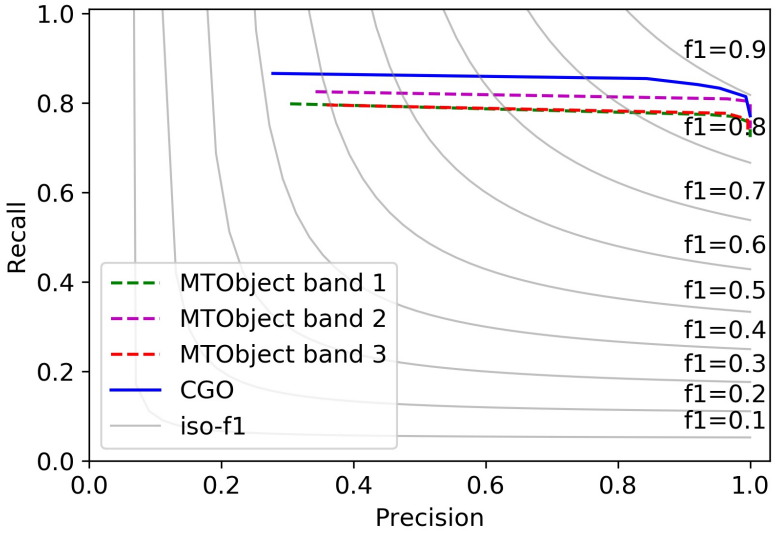


Figure 4.13: Precision-recall curves of source detection in simulated 3-band astronomical images. The proposed multiband detection method based on the component graph (CGO) is compared to the reference single band method based on a max tree (MTObject).

accurately account for all the fine aspects of real observations which can induce a large bias in the parameter estimation.

To solve this issue we proposed to use a connected filter in order to detect and mask the brightest parts of the galaxies, called H II regions, that can not be accurately described by the models. H II regions correspond to ionized hydrogen clouds which are produced by supernova, *i.e.*, the explosive death of short-lived massive stars. Such stars appear mostly along the spiral arms of the galaxy where hydrogen clouds tend to collapse into new stars. We have identified 3 features to characterize H II regions:

- *energy*: the surface brightness of the region is significantly larger than the noise energy;
- *size*: for physical reasons, the size of these regions is limited, they cannot be neither too small nor too large; and
- *colour*: H II regions are significantly brighter in short wavelengths, they are blueish.

In order to represent the astronomical multiband images with a max tree, we proposed a total vectorial order of the pixel values motivated by physical considerations. This order is based on the combination of several techniques [57]: a reduced order defined as a normalized truncated energy function, a quantified and normalized order and finally, a lexicographic order. Formally, let $v, v' \in \mathbb{R}^n$ be two pixel values, we define

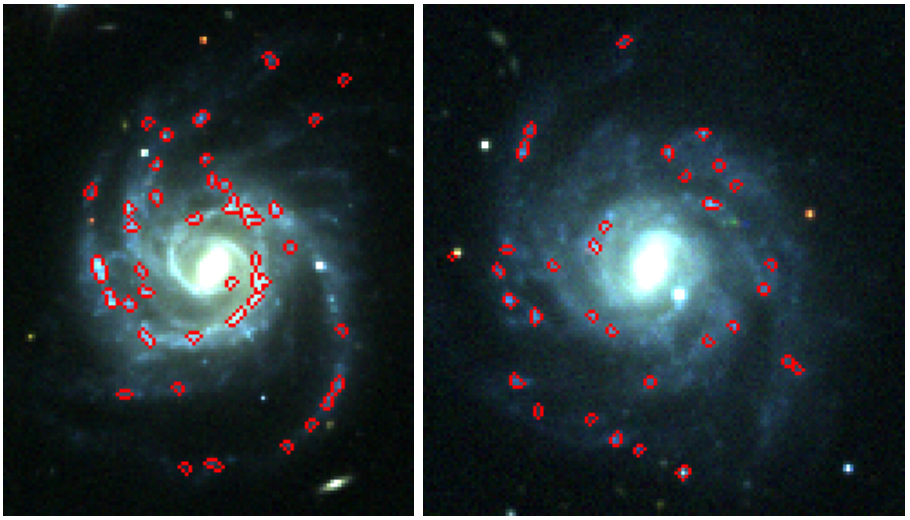


Figure 4.14: H II regions detection in two 5-bands astronomical images of spiral galaxies. As expected, the detected regions are mostly located along the spiral arms of the galaxies.

the pre-order $\leq_{Ap}$ by:

$$v \leq_{Ap} v' \Leftrightarrow \left[\lfloor \mathrm{E}_n(v) \rfloor, \left\lfloor \frac{v_1}{k\sigma_1} \right\rfloor, \ldots, \left\lfloor \frac{v_n}{k\sigma_n} \right\rfloor\right] \leq_L \left[\lfloor \mathrm{E}_n(v') \rfloor, \left\lfloor \frac{v'_1}{k\sigma_1} \right\rfloor, \ldots, \left\lfloor \frac{v'_n}{k\sigma_n} \right\rfloor\right] \tag{4.1}$$

where $\leq_L$ is the lexicographic order, $\sigma_1, \ldots, \sigma_n$ are the standard deviation of the noise in the respective bands, k is a confidence factor ($k = 3$), $\lfloor \cdot \rfloor$ is the floor function. Moreover, the bands are sorted by the size of the point spread function (blurry bands at last). $\mathrm{E}_n(v)$ is the normalized energy defined as: $\mathrm{E}_n(v) = \left\| \frac{v_1}{k\sigma_1}, \ldots, \frac{v_n}{k\sigma_n} \right\|$. To obtain a total order, we extend $\leq_{Ap}$ with a lexicographic order applied to the initial spectral bands.

Based on this particular vectorial ordering and the features of H II regions, we designed [21, 37] a custom multiband connected filter to detect these regions (see Figure 4.14). We showed [14,21] that the introduction of this detector to dynamically mask the H II regions detected along the spiral arms of a galaxy enables to significantly improve the quality of the estimation of the model parameters (see Figure 4.15).

4.4 Open source library for hierarchical graph analysis: Higra

Software and in particular open-source software plays a key-role in nowadays science. It is a prominent ingredient of successful dissemination of research. It does not only allow users, students and researchers to easily use and further develop the proposed method but also greatly contributes to the reproducibility of research.

In this context, we contributed by developing an open-source library called Higra [7]:

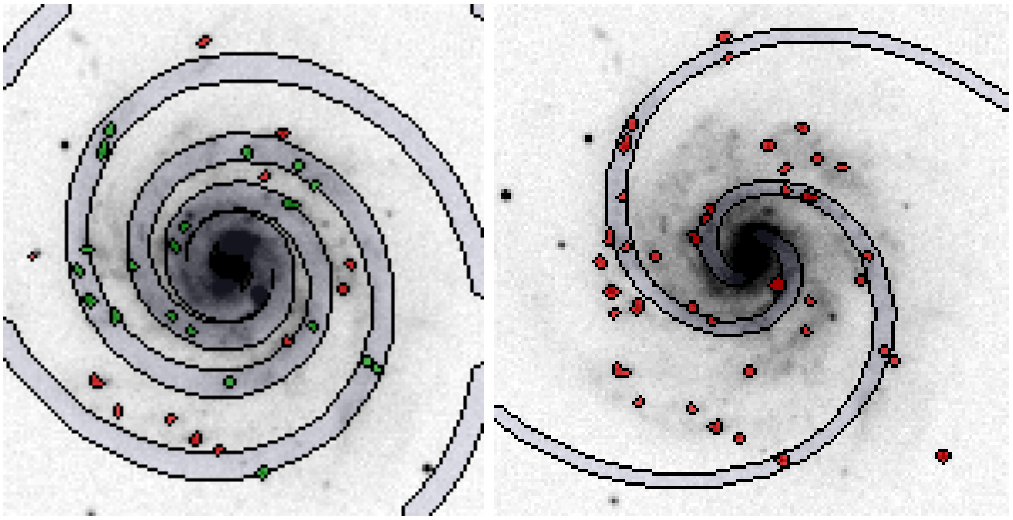


Figure 4.15: From left to right: shape of the estimated spiral arms on the galaxy PGC 35538 with and without adaptive masking of detected H II regions. Red and green regions correspond to the H II regions detected by the proposed mutiband connected filter. With adaptive masking, green regions are masked while red regions which are too far from the currently estimated spiral arms (gray area) are not masked. The adaptive masking of H II regions attracts the spiral arms towards those regions and helps to get a better estimate of their parameters.

Hierarchical Graph Analysis[1]. Higra is a Python library with a C++ back-end for efficient sparse graph analysis with a special focus on hierarchical methods. It aims at providing standard and state-of-the-art algorithms for hierarchical graph analysis capable of handling large amount of data (up to dozen of millions of vertices on a classical desktop computer). It is a generic toolbox for hierarchical graph representation construction, processing and assessment: it is thus not focused toward a particular application or domain and its fundamental functions can be used in a variety of situations.

Higra provides a Python API (Application Programming Interface) to ease its usage and to benefit from the synergies created by the large amount of scientific libraries available in the Python ecosystem. It is available through the official Python package repository *PyPi* which makes it usable from any Python environment, on Linux, MacOS, or Windows, with a single command line. This API is thought to be usable by students and expert researchers willing to quickly develop new applications, experiment new methods, or develop proofs of concepts. Vectorized operations on hierarchical representations enable writing various algorithms working on hierarchical representations efficiently in Python. It is also thought for seamless integration with classical Python data analysis pipelines such as *Numpy*, *Scikit-Learn* and modern optimization framework such as *PyTorch* and *TensorFlow*. The Python interface is backed-up by a C++ module where core algorithms are implemented to ensure high performances. The C++ module is also usable as a standalone library since it does not have any dependency to the Python runtime.

[1] `https://github.com/higra/Higra`

Higra tries to follow the best practice of software engineering:

- it is thoroughly tested (unit test coverage of 99%);
- its application programming interface is well documented[2]. This documentation is complemented by a set of *Jupyter Notebooks* showing how to use the library and serving as online demonstrations[3] (see Figure 4.16); and
- it's development is driven by a continuous integration pipeline (each code modification is automatically tested on all supported platforms before being merged into the main repository) and a continuous delivery pipeline (pushing a new tag to the repository automatically compiles and releases a new version of the library on all supported platforms, the online documentation is automatically updated).

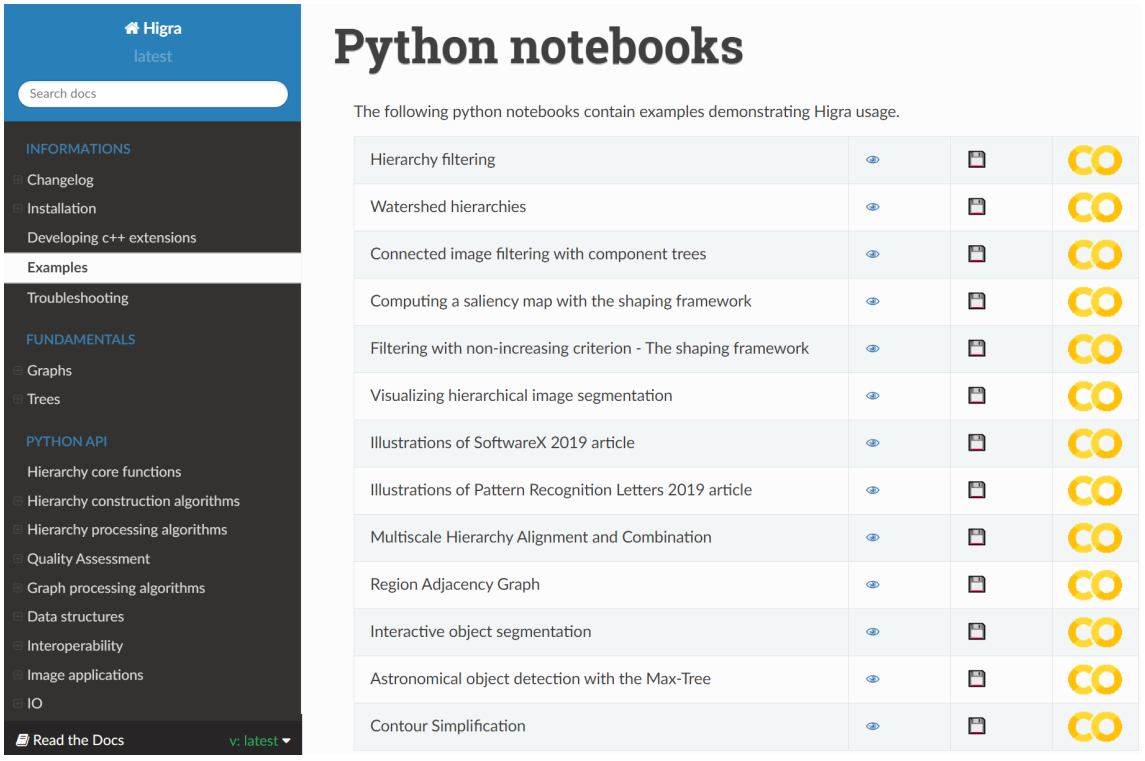


Figure 4.16: Screen-shot of the online documentation of the Higra library showing the list of currently available Python demonstration notebooks. All the notebooks can be executed online in Jupyter like environments such as Google Colab.

4.4.1 Main functions

Higra contains a large amount of classical and recent algorithms for the construction, the manipulation, and the analysis of hierarchical graph representations:

- **efficient methods and data structures to handle the dual representations of hierarchical clustering**: trees [88] (dendrograms) and saliency

[2] `https://higra.readthedocs.io`
[3] `https://higra.readthedocs.io/en/stable/notebooks.html`

maps [100] (ultrametric distances);

- **hierarchical clustering**: quasi-flat-zone hierarchies [164], watershed hierarchies [99, 161], agglomerative clustering [177] (single-linkage [32, 127], average-linkage, complete-linkage, exponential-linkage [269], Ward [252], or user-provided linkage rule), constrained connectivity hierarchies [236];
- **component trees**: min and max trees [133, 224];
- **manipulate and explore hierarchies**: simplification [9, 22], accumulators, cluster extraction, various attributes [273] (size, volume, dynamics, perimeter, compactness, moments, etc.), horizontal and non-horizontal cuts, alignment of hierarchies [201];
- **optimization on hierarchies**: optimal cuts, energy hierarchies [128, 138];
- **algorithms on graphs**: accumulators, vertices and clusters dissimilarities, region adjacency graphs, minimum spanning trees and forests, (seeded) watershed cuts [97];
- **assessment**: supervised assessment of graph clusterings and hierarchical clusterings [10, 91, 129];
- **image toolbox**: special methods for grid graphs, tree of shapes [123], multivariate tree of shapes [83,86], multi-scale combinatorial grouping [148], optimization of Mumford-Shah energy [176].

4.4.2 Example in image filtering

The example in Figure 4.17 demonstrates the use of hierarchical clustering for image filtering. The strategy followed here is to first construct a watershed hierarchy by area [99, 161] of the gradient of the image represented as an edge-weighted graph. Then, a flat clustering containing k clusters is extracted from the hierarchical representation. Finally, the color of each pixel contained in a cluster is defined as the mean color, in the original image, of the pixels inside the cluster.

A 4-adjacency edge weighted graph is built from the gradient of an image on lines 5 and 6. Then a watershed hierarchy by area of the graph is constructed on line 9. The saliency map of the hierarchy, which weights each edge of the graph by the ultrametric distance between its extremities, is computed for illustrative purpose on line 10 and is plotted in the 2D Khalimsky grid. Then, the mean image colour inside each region of the hierarchy is computed on line 16. The object of the class `HorizontalCutExplorer`, instantiated on line 19, eases the construction of horizontal-cuts (flat clustering) of the given hierarchy. It is used to extract several cuts containing different number of regions (line 19) and the images corresponding to these cuts are reconstructed using the mean image color of their regions (line 20).

```
image    = imread("101087.jpg") / 255
gradient = imread("101087_SED.png") / 255

# Edge weighted 4-adjacency graph
graph        = get_4_adjacency_graph(gradient.shape[:2])
edge_weights = weight_graph(graph, gradient, WeightFunction.mean)

# Watershed hierarchy by area and its saliency map
tree, altitudes = watershed_hierarchy_by_area(graph, edge_weights)
sm              = saliency(tree, altitudes)
imshow(image); imshow(1 - gradient)
imshow(1 - graph_4_adjacency_2_khalimsky(graph, sm) ** 0.5)

# Get horizontal cuts containing different number of regions
# and colorize them with the mean pixel values inside each region
mean_color = attribute_mean_weights(tree, image)
cut_helper = HorizontalCutExplorer(tree, altitudes)
for c in [25, 50, 100]:
    cut        = cut_helper.horizontal_cut_from_num_regions(c)
    simplified = cut.reconstruct_leaf_data(tree, mean_color)
    imshow(simplified)
```

Figure 4.17: Example of image filtering with the proposed library Higra: image simplification with a watershed hierarchy. From left to right: original image, gradient, saliency map of the watershed hierarchy by area of the gradient, simplified image reconstructed from the hierarchy with respectively 25, 50, and 100 regions.

Chapter 5

Conclusion and future research directions

5.1 Conclusion

This manuscript summarized our main research contributions in the area of hierarchical image representations and related analysis methods. These contributions cover the whole range of scientific developments, from theory to applications:

- the theoretical works on the axiomatic of connections led to the proposal of novel connected operators that proved useful in astronomical and document image analysis (Sections 2.1 and 2.2);
- we generalized the theory of set connections to directed graphs, proposed efficient algorithms for this new case and showed its potential on medical imaging (Section 2.3);
- our theoretical analysis of the common hierarchies of segmentations used in mathematical morphology allowed us to better understand how those representations relate to each others and to propose novel and efficient algorithms for constructing and for manipulating them (Sections 3.1 and 3.2);
- those theoretical links and these efficient algorithms also allowed us to propose a novel approach to optimize hierarchical segmentations in a very versatile framework based on gradient descent method (Section 3.3);
- we proposed a novel supervised framework to assess hierarchies of segmentations in the context of natural image analysis which allowed us to better understand the strengths and weaknesses of different hierarchical segmentation methods and to optimize the hyper-parameters of the considered methods (Section 4.1);
- we developed specific solutions based on connected operators and hierarchies of segmentations to solve several practical problems in collaborative projects with the cosmetic industries on automatic skin aging characterization (Section 4.2) and with cosmologists on automatic astronomical image analysis (Section 4.3); and finally,

- we implemented most of the proposed methods concerning hierarchical graph analysis into a novel open source Python library making our developments easily accessible to anyone (Section 4.4).

While most of our research works fit well into the topic of this manuscript, we also developed other research projects which are not presented here but which surely contributed to our views on research. Those works are related to:

- The hit-or-miss transform: a non-linear operator based on mathematical morphology, used to detect objects in images. The hit-or-miss operator relies on two structuring elements, one that must fit inside the object and one that must fit outside, to detect potential matches. We contributed to improve the robustness against noise of this operator [12,39] and to propose novel definitions for color image analysis [1, 26].
- Monte Carlo optimization for non convex problems: we developed a direct model for complex multiband observations of galaxies and proposed a method based on Monte Carlo methods and simulated annealing to inverse the model on real images [14, 19, 40].
- Conditional random fields for 3d image segmentation: we proposed a biologically motivated conditional random field model to automatically segment the dermis and the epidermis in confocal reflectance microscopy images of the skin [16, 43].

5.2 Research project

While this manuscript does not follow a chronological order, some general tendencies in the evolution of the research projects we want to promote may be foreseeable. First concerning the field of applications, while image analysis has been at the heart of our research activity, most of the developed methods work on graphs and could thus be applied to other data. Second, machine learning approaches have led to tremendous progress in various research fields; those methods are efficient but struggle to provide guaranties on their results and we believe that combining the expertize we have on hierarchical representations with such learning methods can lead to powerful structured learning methods. This is the fundamental idea of our midterm research objective which is to propose methods for the supervised learning of hierarchical representations of graphs. This project, called ULTRA-LEARN, was submitted in 2020 to the young researcher (JCJC) call of the French national research agency and was granted a four year funding.

This research project will leverage on the preliminary results obtained in [3, 23] where we proposed a generic way to perform gradient descent based optimization of hierarchical loss functions (see Section 3.3).

The method that we want to develop will be:

- **Generic** The developed methods will work on weighted graphs, also known as weighted networks in the literature, *i.e.*, models where the dataset samples are represented by nodes linked by edges and where nodes and edges can be weighted by scalar or vectorial values. Such structure is highly generic, enabling

to model a wide variety of problems arising in various applications fields.

- **Modular** The methods will rely on classical continuous optimization algorithms. Their integration in modern continuous optimization framework will be straightforward. This will ensure that the proposed methods will simultaneously benefit from and enrich the ecosystem developed around those approaches.
- **Scalable** The method will be scalable with respect to the number of nodes in the graph and to the dimension of the weights (features). Our goal is to support at least in the order of dozen of millions of nodes and weights with up to thousands of dimensions.
- **Supervised** Supervised methods in machine learning have benefited from spectacular progress in recent years. The developed method will be able to learn how to construct hierarchical clustering from examples. We will handle the completely supervised case, where examples are themselves hierarchical clusterings and the semi-supervised case, where the examples are only partial information on the hierarchical clustering.

These ideas can be formalized as follows: in the completely supervised case, we are given a set of graphs $\{G_i = (V_i, E_i)\}_{i\in[1,K)}$ with vertex weights $v_i : V_i \mapsto \mathbb{R}^N$ and edges weights $e_i : E_i \mapsto \mathbb{R}^M$, with associated ground-truth hierarchical clustering y_i. The goal is then to devise a method for learning the parameters θ of a hierarchical clustering method m_θ such that $m_\theta(G_i, v_i, e_i)$ is close to y_i. In other words, we will optimize the parameters m_θ in order to minimize the loss between the predicted hierarchical clusterings and the expected ones. This case typically corresponds to the hierarchical image segmentation problem, where one have thousands or millions of images, represented as vertex valued graphs, and the goal is then to obtain a hierarchical segmentation of each image.

In the semi-supervised case, we are given a single graph $G = (V, E)$ with vertex weights $v : V \mapsto \mathbb{R}^N$ and edges weights $e : E \mapsto \mathbb{R}^M$, and some annotations on a subset of the vertices of V. The goal is then to find a hierarchical clustering that best fit the input graph and the partial annotations we have. Those partial annotations will typically be a partial (hierarchical) clustering of some vertices of the graph. This will also lead to optimization problems where one seeks to maximize simultaneously the intrinsic quality of the hierarchical clustering and its accordance with the annotations.

This project involves different aspects which can be organized in three work-packages (see Figure 5.1).

Ultrametric network: The objective of the first work package is to devise the models that will produce a hierarchical clustering of the input data and whose parameters will be optimized in an end-to-end manner. This work package is divided into two tasks:

1. *Definition of ultrametric layers:* The objective of this task is to propose and study ultrametric layers, *i.e.*, operators associating to any edge weighted graph an ultrametric which is differentiable with respect to the edge weights of the

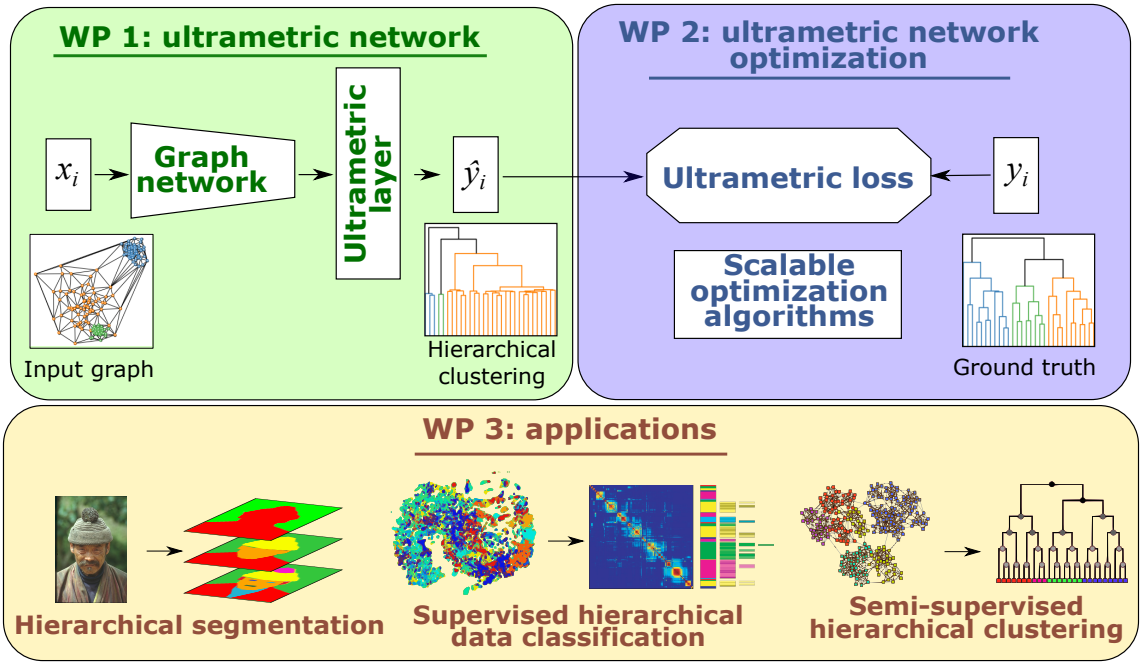


Figure 5.1: Organization of our future research project on supervised hierarchical clustering.

input graph. In other words, it is a hierarchical clustering algorithm through which we can back-propagate a gradient on the edge weights of the graph. There have been many hierarchical clustering algorithms proposed over the years and we will first study their applicability as ultrametric layers. One of the main selection criterion is their differentiability as none of the algorithms were created with this feature in mind.

2. *Dissimilarity learning for ultrametric layers:* The objective of this task is to propose and study networks designed to produce graph embeddings into metric spaces adapted for hierarchical clustering. More precisely, given a weighted graph, the objective of this work package is to propose network architectures that will produce a new edge weighted graph with the same set of nodes and whose edges are weighted by dissimilarities. Indeed, one of the key element in the success of modern machine learning methods is their ability to automatically learn rich features from raw data using deep networks. In the context of this project we are especially interested in networks designed to perform metric learning, *i.e.*, networks that can estimate the dissimilarity between elements. However, there are nowadays two significantly separated approaches to perform deep learning on graph. The first one, that has popularized deep learning, is based on convolutional layers operating on regular grid graphs (classically adjacency graphs of pixels in image analysis): the natural spatial embedding of the nodes in the Euclidean plane then play a key role in the architecture of the network. The second one acts on arbitrary graphs: the nodes of the graph do not have any natural embedding in the Euclidean space and everything has to be learned from the features. We will address these two cases to propose learnable embeddings that can be used in the context of hierarchical learning.

Ultrametric network optimization: This work package aims at studying the definition of dissimilarity measures on the space of ultrametrics and at developing efficient algorithms to train ultrametric network according to these measures. This work package is divided into two tasks:

1. *Ultrametric loss:* The objective of this task is to develop loss functions for hierarchical clusterings, *i.e.*, differentiable dissimilarity measures between two hierarchical clusterings. The design of efficient and meaningful loss functions is one of the key elements in machine learning. While the primary goal of a loss function is to measure the difference between a prediction and a ground-truth, a good loss function must also satisfy several properties: efficient computation, good gradient propagation, robustness to ground-truth and model noise. There exists currently no standard way for measuring the distance between two hierarchical clusterings. There has been a large amount of works on tree edit distances [70] or Gromov-Wassertstein distances [88], however, these methods are combinatorial and they have a very large time complexity. We will address two main cases: 1) graphs benefiting from a natural node embedding in the Euclidean space such as pixels graphs, and 2) general graphs.
2. *Scalable optimization for ultrametric networks:* The objective of this task is to propose scalable algorithms to optimize ultrametric network, *i.e.*, the composition of the dissimilarity estimation network and of the ultrametric layer. The scalability of the training process of neural networks depends on two key ingredients: 1) the fast evaluation of the forward and backward paths of the network; and 2) the possibility to perform a training step without considering all data at once. The fast evaluation of traditional neural network has been made possible by specialized hardware relying on massive parallelism such as Graphical Processing Unit (GPU) or Tensor Processing Unit (TPU). On the contrary, the second ingredient is given by a specific algorithmic solution: the mini-batch approach in stochastic gradient descent. We will study these two aspects in the context of the project.

Applications: This work package aims at grounding the developments of the two previous work packages with real applications. It is divided into three tasks:

1. *Supervised hierarchical image segmentation:* The objective of this task is to develop, to train and to assess an end-to-end hierarchical segmentation method for natural image analysis. Hierarchical clustering is a classical method in image analysis that is traditionally used to structure and reduce the size of the search space in image segmentation and object detection. Hierarchical clustering is part of state-of-the-art pipelines for image segmentation [121, 148], but they are currently used as a post-processing step rather than trained end-to-end. However, the rich contextual information provided by objects-and-parts approaches are known to improve the precision, the robustness, and the interpretability of vision models [166]: we thus believe that training deep vision models to predict a whole hierarchy of segmentations could lead to significant improvement. In this task, we will develop an end-to-end method for learning an image hierarchical segmentation method.

2. *Supervised hierarchical data classification:* The objective of this task is to propose and train a classifier on data following an ontology/hierarchy of classes. In data classification, one usually tries to predict a single class for each sample. However, in many cases, it is possible to have an ontology of classes, *i.e.*, a hierarchy of classes. For example many real world classification tasks in computer vision involves the WordNet database which can be seen as a lexical ontology. In such ontology, the *cat* and the *dog* classes are indeed subclasses of the *mammal* class and a *cat* is closer to a *dog* than a *bird*. Classifiers are then trained to predict classes at a given level of the class hierarchy that depends of the target application. In this task, we will explore how to develop a classifier that can take advantage of this known class ontology.
3. *Semi-supervised hierarchical data clustering:* The objective of this task is to develop a hierarchical clustering method that can take advantage of a hierarchical prior on some of the elements of the dataset. Clustering is often an ambiguous task and several authors have suggested that a limited amount of supervision can lead to a large improvement in results [60, 198]. In such semi-supervised clustering, the algorithm is given a set of data with a few annotated samples which are sometimes called seeds or markers. Usually, these annotations conveys only flat clustering information: given two annotated samples we can say if they are in the same cluster or not. In real applications these seeds or markers are usually given interactively by an expert who can refine the result of the algorithm by providing new information. We will study how such semi-supervision can help to devise better hierarchical clustering algorithm.

Chapter 6

Bibliography

Submitted journal manuscripts

[1] F. Macfarlane, P. Murray, S. Marshall, B. Perret, A. Evans, and H. White. Robust object detection in colour images using a multivariate percentage occupancy hit-or-miss transform. Journal article, submitted, 2020.

[2] T. Xuan Nguyen, G. Chierchia, L. Najman, and B. Perret. CGO: Multiband astronomical source detection with component-graphs. Journal article, submitted, 2020.

Journal articles

[3] G. Chierchia and B. Perret. Ultrametric fitting by gradient descent. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124004, 2020.

[4] D.S. Maia, J. Cousty, L. Najman, and B. Perret. Properties of combinations of hierarchical watersheds. *Pattern Recognition Letters*, 128:513–520, 2019.

[5] D.S. Maia, J. Cousty, L. Najman, and B. Perret. Characterization of graph-based hierarchical watersheds: theory and algorithms. *Journal of Mathematical Imaging and Vision*, pages 1–32, 2020.

[6] B. Perret. Inf-structuring Functions: A Unifying Theory of Connections and Connected Operators. *Journal of Mathematical Imaging and Vision*, 51(1):171–194, 2015.

[7] B. Perret, G. Chierchia, J. Cousty, S.J.F. Guimarães, Y. Kenmochi, and L. Najman. Higra: Hierarchical graph analysis. *SoftwareX*, 10:100335, 2019.

[8] B. Perret and Ch. Collet. Connected image processing with multivariate attributes: an unsupervised Markovian classification approach. *Computer Vision and Image Understanding*, 133:1–14, 2015.

[9] B. Perret, J. Cousty, S.J.F. Guimarães, Y. Kenmochi, and L. Najman. Removing non-significant regions in hierarchical clustering and segmentation. *Pattern*

Recognition Letters, 128:433–439, 2019.

[10] B. Perret, J. Cousty, S.J.F. Guimarães, and D.S. Maia. Evaluation of hierarchical watersheds. *IEEE Transactions on Image Processing*, 27(4):1676–1688, 2018.

[11] B. Perret, J. Cousty, O. Tankyevych, H. Talbot, and N. Passat. Directed connected operators: Asymmetric hierarchies for image filtering and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(6):1162–1176, 2015.

[12] B. Perret, S. Lefèvre, and Ch. Collet. A Robust Hit-or-Miss Transform for Template Matching in Very Noisy Astronomical Images. *Pattern Recognition*, 42(11):2470–2480, November 2009.

[13] B. Perret, S. Lefèvre, Ch. Collet, and E. Slezak. Hyperconnections and Hierarchical Representations for Grayscale and Multiband Image Processing. *IEEE Transactions on Image Processing*, 21(1):14–27, 2012.

[14] B. Perret, M. Vincent, Ch. Collet, and E. Slezak. Hierarchical multispectral galaxy decomposition using a MCMC algorithm with multiple temperature simulated annealing. *Pattern Recognition*, 44(6):1328–1342, 2011.

[15] J. Robic, A. Nkengne, B. Perret, M. Couprie, H. Talbot, G. Pellacani, and K. Vie. Clinical validation of a computer-based approach for the quantification of the skin aging process of women using in-vivo confocal microscopy. *Journal of the European Academy of Dermatology and Venereology*, page to appear, 2020.

[16] J. Robic, B. Perret, A. Nkengne, M. Couprie, and H. Talbot. Three-dimensional conditional random field for the dermal–epidermal junction segmentation. *Journal of Medical Imaging*, 6(02):1, April 2019.

[17] B. Vollmer, B. Perret, M. Petremand, F. Lavigne, Ch. Collet, W. Van Driel, F. Bonnarel, M. Louys, S. Sabatini, and L.A. MacArthur. Simultaneous multiband detection of Low Surface Brightness galaxies with Markovian modelling. *Astronomical Journal*, 145(2):36, 2013.

Books, book chapters, and dissertations

[18] *Mathematical Morphology and Its Applications to Signal and Image Processing*, volume 11564 of *Lecture Notes in Computer Science*. Springer Nature Switzerland AG, 2019.

[19] Ch. Collet, B. Perret, and M. Vincent. Panoramic Integral-Field Spectrograph: Ultraspectral Data to Understand the History of the Universe. In Ch. Collet, J. Chanussot, and K. Chehdi, editors, *Digital Signal and Image Processing Series*, Multivariate Image Processing, pages 437–450. ISTE Ltd and John Wiley & Sons Inc, 2009.

[20] S. Lefèvre, E. Aptoula, B. Perret, and J. Weber. Morphological Template

Matching in Color Images. In *Advances in Low-Level Color Image Processing*, volume 11 of *Lecture Notes in Computational Vision and Biomechanics*, pages 241–277. Springer Netherlands, 2014.

[21] B. Perret. *Multispectral galaxy characterization with a hierarchy of model and connected component trees.* Theses, Université de Strasbourg, November 2010.

Conferences

[22] I.B. Barcelos, G.B. da Fonseca, L. Najman, Y. Kenmochi, B. Perret, J. Cousty, Z. KG do Patrocínio, and S.J.F. Guimarães. Exploring hierarchy simplification for non-significant region removal. In *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 100–107. IEEE, 2019.

[23] G. Chierchia and B. Perret. Ultrametric fitting by gradient descent. In *Advances in neural information processing systems*, pages 3181–3192, 2019.

[24] J. Cousty, L. Najman, and B. Perret. Constructive links between some morphological hierarchies on edge-weighted graphs. In *International Symposium on Mathematical Morphology*, volume 7883 of *Lecture Notes in Computer Science*, pages 85–96, Uppsala, Sweden, May 2013. Springer.

[25] M. Louys, B. Perret, B. Vollmer, F. Bonnarel, S. Lefèvre, and C. Collet. LSB Galaxies Detection Using Markovian Segmentation on Astronomical Images. In *Astronomical Data Analysis Software and Systems XVII*, volume 394 of *Astronomical Society of the Pacific Conference Series*, pages 125–128, August 2008.

[26] F. Macfarlane, P. Murray, S. Marshall, B. Perret, A. Evans, and H. White. A colour hit-or-miss transform based on a rank ordered distance measure. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 588–592, Sep. 2018.

[27] D.S. Maia, A. De A. Araujo, J. Cousty, L. Najman, B. Perret, and H. Talbot. Evaluation of combinations of watershed hierarchies. In *International Symposium on Mathematical Morphology*, Fontainebleau, France, May 2017.

[28] D.S. Maia, J. Cousty, L. Najman, and B. Perret. On the probabilities of hierarchical watersheds. In *International Symposium on Mathematical Morphology*, Mathematical Morphology and Its Applications to Signal and Image, Saarbrücken, Germany, July 2019.

[29] D.S. Maia, J. Cousty, L. Najman, and B. Perret. Recognizing hierarchical watersheds. In *International Conference on Discrete Geometry for Computer Imagery*, pages 300–313. Springer, 2019.

[30] D.S. Maia, J. Cousty, L. Najman, and B. Perret. Watersheding hierarchies. In *International Symposium on Mathematical Morphology*, Lecture Notes on Computer Science, Saarbrücken, Germany, July 2019.

[31] D.S. Maia, J. Cousty, L. Najman, and B. Perret. Watersheding hierarchies. In

Mathematical Morphology and Its Applications to Signal and Image Processing, pages 124–136, Cham, 2019. Springer International Publishing.

[32] L. Najman, J. Cousty, and B. Perret. Playing with Kruskal: algorithms for morphological trees in edge-weighted graphs. In *International Symposium on Mathematical Morphology*, volume 7883 of *Lecture Notes in Computer Science*, pages 135–146, Uppsala, Sweden, May 2013. Springer.

[33] F.T.L.R. Nhimi, Z. Patrocínio, B. Perret, J. Cousty, and S.J.F. Guimarães. Evaluation of morphological hierarchies for supervised video segmentation. In *the 33rd Annual ACM Symposium on Applied Computing*, Pau, France, April 2018. ACM Press.

[34] K. Otiniano-Rodríguez, A. De A. Araújo, G. Cámara-Chávez, J. Cousty, S.J.F. Guimarães, and B. Perret. Hierarchy-Based Salient Regions: A Region Detector Based on Hierarchies of Partitions. In *CIARP 2018: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 444–452. March 2019.

[35] B. Perret, J. Cousty, J.C. Rivera Ura, and S.J.F. Guimarães. Evaluation of morphological hierarchies for supervised segmentation. In *International Symposium on Mathematical Morphology*, volume 9082 of *Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 39–50, Reykjavik, Iceland, May 2015. Springer.

[36] B. Perret, S. Lefèvre, and Ch. Collet. Toward a new axiomatic for hyperconnections. In *International Symposium on Mathematical Morphology*, pages 85–95, Italy, 2011.

[37] B. Perret, S. Lefèvre, Ch. Collet, and E. Slezak. Connected component trees for multivariate image processing applications in astronomy. In *20th IAPR International Conference on Pattern Recognition (ICPR)*, pages 4089–4092, Turkey, 2010.

[38] B. Perret, S. Lefèvre, Ch. Collet, and E. Slezak. From hyperconnections to hypercomponent tree: Application to document image binarization. In *Workshop on Applications of Digital Geometry and Mathematical Morphology (WADGMM)*, Turkey, 2010.

[39] B. Perret, S. Lefèvre, Ch. Collet, and B. Vollmer. Astronomical Object Detection with a Robust Hit-or-Miss Transform. In *16th EURASIP European Signal Processing Conference (EUSIPCO)*, pages –, Switzerland, 2008.

[40] B. Perret, M. Vincent, Ch. Collet, and E. Slezak. Galaxy Decomposition in Multispectral Images Using Markov Chain Monte Carlo Algorithms. In *Scandinavian Conference on Image Analysis*, volume 5575 of *Lecture Notes in Computer Science*, pages 209–218, Oslo, Norway, June 2009. Springer.

[41] Benjamin Perret. Inf-structuring functions and self-dual marked flattenings in bi-heyting algebra. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 365–376. Springer, 2013.

[42] J. Robic, A. Nkengne, B. Perret, M. Couprie, and H. Talbot. Automated quantification of the epidermal aging process using in-vivo confocal microscopy. In *International Symposium on Biomedical Imaging*, Prague, Czech Republic, April 2016.

[43] J. Robic, B. Perret, A. Nkengne, M. Couprie, and H. Talbot. Classification of the dermal-epidermal junction using in-vivo confocal microscopy. In *IEEE International Symposium on Biomedical Imaging*, pages 252–255, Melbourne, Australia, April 2017.

[44] J. Robic, B. Perret, A. Nkengne, M. Couprie, and H. Talbot. Self-dual pattern spectra for characterising the dermal-epidermal junction in 3D reflectance confocal microscopy imaging. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, volume 11564 of *Lecture Notes in Computer Science*, pages 508–519, Saarbrücken, Germany, 2019. Springer.

[45] M.H.F. Wilkinson, C. Haigh, S. Gazagnes, P. Teeninga, N. Chamba, T. Xuan Nguyen, H. Talbot, L. Najman, B. Perret, G. Chierchia, A. Venhola, and R. Peletier. Sourcerer: A robust, multi-scale source extraction tool suitable for faint anddiffuse objects. In *The Realm of the Low-Surface-Brightness Universe Proceedings IAU Symposium*, number 355. International Astronomical Union, 2019.

[46] T. Xuan Nguyen, G. Chierchia, L. Najman, A. Venhola, C. Haigh, R. Peletier, M.H.F. Wilkinson, H. Talbot, and B. Perret. CGO: Multiband astronomical source detection with component-graphs. In *International Conference on Image Processing (ICIP)*, United Arab Emirates, 2020.

Advised PhD dissertations

[47] D.S. Maia. *A study of hierarchical watersheds on graphs with applications to image segmentation.* PhD thesis, Paris Est, 2019.

[48] J. Robic. *Automated characterization of skin aging using in vivo confocal microscopy.* PhD thesis, Paris Est, 2018.

Chapter 7

External references

[49] W. Ackermann. Zum hilbertschen aufbau der reellen zahlen. *Mathematische Annalen*, 99(1):118–133, 1928.

[50] N. Ailon and M. Charikar. Fitting tree metrics: Hierarchical clustering and phylogeny. *SIAM Journal on Computing*, 40(5):1275–1291, 2011.

[51] M.A. Aizerman. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25:821–837, 1964.

[52] S. Alpert, M. Galun, A. Brandt, and R. Basri. Image segmentation by probabilistic bottom-up aggregation and cue integration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):315–327, 2011.

[53] S. Alsaody and J. Serra. Connective segmentation generalized to arbitrary complete lattices. In *Mathematical Morphology and Its Applications to Image and Signal Processing*, volume 6671 of *Lecture Notes in Computer Science*, pages 61–72. Springer, 2011.

[54] J. Angulo and J. Serra. Morphological coding of color images by vector connected filters. In *Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on*, volume 1, pages 69–72, 2003.

[55] Jesus Angulo. Hierarchical laplacian and its spectrum in ultrametric image processing. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 29–40. Springer, 2019.

[56] Jesús Angulo and Santiago Velasco-Forero. Morphological semigroups and scale-spaces on ultrametric spaces. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 28–39. Springer, 2017.

[57] E. Aptoula and S. Lefèvre. A comparative study on multivariate mathematical morphology. *Pattern Recognition*, 40(11):2914–2929, 2007.

[58] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011.

[59] M. Babai, A.S. Chowdhury, and M.H.F. Wilkinson. A graph formalism for time and memory efficient morphological attribute-space connected filters. In *Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 281–294. Springer International Publishing, 2019.

[60] E. Bair. Semi-supervised clustering methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 5(5):349–361, 2013.

[61] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine learning*, 56(1-3):89–113, 2004.

[62] M.A. Bender and M. Farach-Colton. The lca problem revisited. In *Latin American Symposium on Theoretical Informatics*, pages 88–94. Springer, 2000.

[63] M.A. Bender, J.T. Fineman, S. Gilbert, and R.E. Tarjan. A new approach to incremental cycle detection and related problems. *ACM Transactions on Algorithms (TALG)*, 12(2):1–22, 2015.

[64] Ch. Berger, T. Geraud, R. Levillain, N. Widynski, A. Baillard, and E. Bertin. Effective component tree computation with application to pattern recognition in astronomical imaging. In *IEEE International Conference on Image Processing*, volume 4, pages 41–44, 2007.

[65] E. Bertin and S. Arnouts. SExtractor: Software for source extraction. *Astronomy and Astrophysics Supplement*, 117:393–404, 1996.

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

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

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

[69] A. Bieniek and A. Moga. A connected component approach to the watershed segmentation. *Computational Imaging and Vision*, 12:215–222, 1998.

[70] Ph. Bille. A survey on tree edit distance and related problems. *Theoretical computer science*, 337(1-3):217–239, 2005.

[71] J. Bilsland, M. Rigby, L. Young, and S. Harper. A rapid method for semi-quantitative analysis of neurite outgrowth from chick DRG explants using image analysis. *Journal of Neuroscience Methods*, 92:75–85, 1999.

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

[73] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[74] C.A. Bouman and M. Shapiro. A multiscale random field model for bayesian image segmentation. *IEEE Transactions on Image Processing*, 3:162–177, 1994.

[75] Y. Boykov and G. Funka-Lea. Graph cuts and efficient N-D image segmentation. *International Journal of Computer Vision*, 70:109–131, 2006.

[76] U. Braga-Neto. Multiscale connected operators. *Journal of Mathematical Imaging and Vision*, 22(2-3):199–216, 2005.

[77] U. Braga-Neto and J. Goutsias. Connectivity on complete lattices: New results. *Computer Vision and Image Understanding*, 85(1):22–53, 2002.

[78] U. Braga-Neto and J. Goutsias. A multiscale approach to connectivity. *Computer Vision and Image Understanding*, 89(1):70–107, 2003.

[79] U. Braga-Neto and J. Goutsias. A theoretical tour of connectivity in image processing and analysis. *Journal of Mathematical Imaging and Vision*, 19(1):5–31, 2003.

[80] U. Braga-Neto and J. Goutsias. Constructing multiscale connectivities. *Computer Vision and Image Understanding*, 99(1):126–150, 2005.

[81] U. Braga-Neto and J. Goutsias. Object-based image analysis using multiscale connectivity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):892–907, 2005.

[82] Michel C. and Gilles B. Topological gray-scale watershed transformation. In *Vision Geometry Vi*, volume 3168, pages 136–146. International Society for Optics and Photonics, 1997.

[83] E. Carlinet. *A Tree of shapes for multivariate images*. PhD thesis, Université Paris-Est, 2015.

[84] E. Carlinet and T. Géraud. A comparison of many max-tree computation algorithms. In *Mathematical Morphology and Its Applications to Signal and Image Processing*, volume 7883 of *Lecture Notes in Computer Science*, pages 73–85. Springer, 2013.

[85] E. Carlinet and Th. Géraud. A comparative review of component tree computation algorithms. *IEEE Transactions on Image Processing*, 23(9):3885–3895, 2014.

[86] E. Carlinet and Th. Géraud. Mtos: A tree of shapes for multivariate images. *IEEE Transactions on Image Processing*, 24(12):5330–5342, 2015.

[87] E. Carlinet and Th. Géraud. Introducing multivariate connected openings and closings. In *Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 215–227. Springer International Publishing, 2019.

[88] G. Carlsson and F. MÃŠmoli. Characterization, stability and convergence of hierarchical clustering methods. *Journal of machine learning research*, 11(Apr):1425–1470, 2010.

[89] M. Charikar and V. Chatziafratis. Approximate hierarchical clustering via sparsest cut and spreading metrics. In *Proceedings of the Twenty-Eighth Annual*

ACM-SIAM Symposium on Discrete Algorithms, pages 841–854. SIAM, 2017.

[90] M. Charikar, V. Chatziafratis, and R. Niazadeh. Hierarchical clustering better than average-linkage. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2291–2304. SIAM, 2019.

[91] B. Charpentier and T. Bonald. Tree sampling divergence: An information-theoretic metric for hierarchical graph clustering. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2067–2073, 2019.

[92] J.M. Chassery. Connectivity and consecutivity in digital pictures. *Computer Graphics and Image Processing*, 9:294—-300, 1979.

[93] V. Chatziafratis, R. Niazadeh, and M. Charikar. Hierarchical clustering with structural constraints. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 774–783, 2018.

[94] V. Cohen-Addad, V. Kanade, and F. Mallmann-Trenn. Hierarchical clustering beyond the worst-case. In *Advances in Neural Information Processing Systems*, pages 6201–6209, 2017.

[95] V. Cohen-Addad, V. Kanade, F. Mallmann-Trenn, and C. Mathieu. Hierarchical clustering: Objective functions and algorithms. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 378–397. SIAM, 2018.

[96] V. Cohen-Addad, V. Kanade, F. Mallmann-Trenn, and C. Mathieu. Hierarchical clustering: Objective functions and algorithms. *Journal of the ACM (JACM)*, 66(4):1–42, 2019.

[97] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts: Minimum spanning forests and the drop of water principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1362–1374, 2008.

[98] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts: Thinnings, shortest path forests, and topological watersheds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):925–939, 2009.

[99] J. Cousty and L. 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.

[100] J. Cousty, L. Najman, Y. Kenmochi, and S. 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.

[101] J. Cousty, L. Najman, and J. Serra. Raising in watershed lattices. In *2008 15th IEEE International Conference on Image Processing*, pages 2196–2199, 2008.

[102] J. Crespo. *Morphological Connected Filters and Intra-region Smoothing for Image Segmentation*. PhD thesis, Georgia Institute of Technology, 1993. UMI Order No. GAX94-15632.

[103] J. Crespo and V. Maojo. The strong property of morphological connected alternated filters. *Journal of Mathematical Imaging and Vision*, 32(3):251–263, 2008.

[104] J. Crespo and R.W. Schafer. Locality and adjacency stability constraints for morphological connected operators. *Journal of Mathematical Imaging and Vision*, 7(1):85–102, 1997.

[105] M. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone. Morphological attribute profiles for the analysis of very high resolution images. *IEEE Transactions on Geoscience and Remote Sensing*, 48(10):3747–3762, 2010.

[106] M. Dalla Mura, J.A. Benediktsson, and L. Bruzzone. Self-dual attribute profiles for the analysis of remote sensing images. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 320–330. Springer, 2011.

[107] M. Dalla Mura, J.A. Benediktsson, B. Waske, and L. Bruzzone. Extended profiles with morphological attribute filters for the analysis of hyperspectral data. *International Journal of Remote Sensing*, 31(22):5975–i5991, 2010.

[108] M. Dalla Mura, A. Villa, J. A. Benediktsson, J. Chanussot, and L. Bruzzone. Classification of hyperspectral images by using extended morphological attribute profiles and independent component analysis. *IEEE Geoscience and Remote Sensing Letters*, 8(3):542–546, 2011.

[109] S. Dasgupta. A cost function for similarity-based hierarchical clustering. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 118–127, 2016.

[110] R. C. de Amorim. Feature relevance in ward's hierarchical clustering using the lp norm. *Journal of Classification*, 32(1):46–62, 2015.

[111] E.D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2-3):172–187, 2006.

[112] M. Di Summa, D. Pritchard, and L. Sanità. Finding the closest ultrametric. *Discrete Applied Mathematics*, 180(10):70–80, 2015.

[113] H Digabel and Christian Lantuéjoul. Iterative algorithms. In *Proc. 2nd European Symp. Quantitative Analysis of Microstructures in Material Science, Biology and Medicine*, volume 19, page 8, 1978.

[114] P. Dollár and C.L. Zitnick. Fast edge detection using structured forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1558–1570, 2014.

[115] 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.

[116] A.X. Falcão, J. Stolfi, and R. de Alencar Lotufo. The image foresting transform:

Theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):19–29, 2004.

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

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

[119] K. Florek, J. Łukaszewicz, J. Perkal, H. Steinhaus, and S. Zubrzycki. Sur la liaison et la division des points d'un ensemble fini. In *Colloquium mathematicum*, volume 2, pages 282–285, 1951.

[120] A. F. Frangi, W. J. Niessen, R. M. Hoogeveen, T. van Walsum, and M. A. Viergever. Model-based quantitation of 3-D magnetic resonance angiographic images. *IEEE Transactions on Medical Imaging*, 18:946–956, 1999.

[121] J. Funke, F. Tschopp, W. Grisaitis, A. Sheridan, C. Singh, S. Saalfeld, and S. C. Turaga. Large scale image segmentation with structured loss based deep learning for connectome reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1669–1680, 2019.

[122] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[123] T. Géraud, E. Carlinet, S. Crozet, and L. Najman. A quasi-linear algorithm to compute the tree of shapes of n-D images. In *Mathematical Morphology and Its Applications to Signal and Image Processing*, volume 7883 of *Lecture Notes in Computer Science*, pages 97–108. Springer, 2013.

[124] D. Gimenez and A.N. Evans. An evaluation of area morphology scale-spaces for colour images. *Computer Vision and Image Understanding*, 110(1):32–42, 2008.

[125] R.C. Gonzalez and R.E. Woods. *Digital Image Processing (3rd Edition)*, chapter 6. Prentice-Hall, Inc., 2006.

[126] M. Götz, G. Cavallaro, Th. Géraud, M. Book, and M. Riedel. Parallel computation of component trees on distributed memory machines. *IEEE Transactions on Parallel and Distributed Systems*, 29(11):2582–2598, 2018.

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

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

[129] K.A. Heller and Z. Ghahramani. Bayesian hierarchical clustering. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, pages 297–304. ACM, 2005.

[130] D. Hoiem, A.A. Efros, and M. Hebert. Recovering occlusion boundaries from an image. *International Journal of Computer Vision*, 91(3):328–346, 2011.

[131] S.L. Horowitz and T. Pavlidis. Picture segmentation by a tree traversal algorithm. *Journal of the ACM (JACM)*, 23(2):368–388, 1976.

[132] M.F. Janowitz. A model for ordinal filtering of digital images. *Statistical Image Processing and Graphics*, 72:25–41, 1986.

[133] R. Jones. Connected filtering and segmentation using component trees. *Computer Vision and Image Understanding*, 75(3):215–228, 1999.

[134] C. Jordan. Nouvelles observations sur les lignes de faıtes et de thalweg. *Comptes rendus des séances de l'académie des sciences*, 75:1023–1025, 1872.

[135] E. Khalimsky. Topological structures in computer science. *International Journal of Stochastic Analysis*, 1(1):25–40, 1987.

[136] E. Khalimsky, R. Kopperman, and P.R. Meyer. Computer graphics and connected topologies on finite ordered sets. *Topology and its Applications*, 36(1):1–17, 1990.

[137] B.R. Kiran and J. Serra. Fusion of ground truths and hierarchies of segmentations. *Pattern Recognition Letters*, 47:63–71, 2014.

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

[139] J.J. Koenderink. The structure of images. *Biological Cybernetics*, 50(5):363–370, 1984.

[140] J.B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

[141] J.-M. Laferté, P. Pérez, and F. Heitz. Discrete markov image modeling and inference on the quadtree. *IEEE Transactions on image processing*, 9(3):390–404, 2000.

[142] F William Lawvere. Intrinsic co-heyting boundaries and the leibniz rule in certain toposes. In *Category theory*, pages 279–281. Springer, 1991.

[143] S. Le Cam, Ch. Collet, and F. Salzenstein. Detection of transient signals in lung sounds: Local approach using a markovian tree with frequency selectivity. *Journal of Signal Processing Systems*, 65(3):445–456, 2011.

[144] B. Leclerc. Description combinatoire des ultramétriques. *Mathématiques et Sciences humaines*, 73:5–37, 1981.

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

[146] C. Longo, A. Casari, B. De Pace, S. Simonazzi, G. Mazzaglia, and G. Pellacani. Proposal for an in vivo histopathologic scoring system for skin aging by means of confocal microscopy. *Skin Research and Technology*, 19(1):e167–e173, 2013.

[147] B. Luo and L. Zhang. Robust autodual morphological profiles for the classification of high-resolution satellite images. *IEEE Transactions on Geoscience and Remote Sensing*, 52(2):1451–1462, 2013.

[148] 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.

[149] P. Maragos. Pattern spectrum and multiscale shape representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):701–716, 1989.

[150] P. Maragos and R.D. Ziff. Threshold superposition in morphological image analysis systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):498–504, 1990.

[151] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *International Conference on Computer Vision*, volume 2, pages 416–423, 2001.

[152] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[153] P. Matas, E. Dokladalova, M. Akil, Th. Grandpierre, L. Najman, M. Poupa, and V. Georgiev. Parallel algorithm for concurrent computation of connected component tree. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 230–241. Springer, 2008.

[154] J.C. Maxwell. L. on hills and dales: To the editors of the philosophical magazine and journal. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 40(269):421–427, 1870.

[155] L.L. McQuitty. Elementary linkage analysis for isolating orthogonal and oblique types and typal relevancies. *Educational and Psychological Measurement*, 17(2):207–229, 1957.

[156] A. Meijster and J.B.T.M. Roerdink. A disjoint set algorithm for the watershed transform. In *9th European Signal Processing Conference (EUSIPCO 1998)*, pages 1–4. IEEE, 1998.

[157] D. Menotti, L. Najman, and A. de Albuquerque Araújo. 1d component tree in linear time and space and its application to gray-level image multithresholding. In *Proceedings of the 8th International Symposium on Mathematical Morphology*, volume 1, pages 437–448, 2007.

[158] F. Meyer. Un algorithme optimal de ligne de partage des eaux. *Actes du 8ème Congrès AFCET*, 2:847–859, 1991.

[159] F. Meyer. Minimum spanning forests for morphological segmentation. In *Mathematical morphology and its applications to image processing*, pages 77–84.

Springer, 1994.

[160] F. Meyer. Topographic distance and watershed lines. *Signal processing*, 38(1):113–125, 1994.

[161] F. Meyer. The dynamics of minima and contours. In *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 329–336. Springer, 1996.

[162] F. Meyer. From connected operators to levelings. In *Mathematical Morphology and its applications to image and signal processing*, pages 191–198. Kluwer, 1998.

[163] F. Meyer. The levelings. In *Mathematical Morphology and its applications to image and signal processing*, pages 199–206. Kluwer, 1998.

[164] F. Meyer and P. Maragos. Morphological scale-space representation with levelings. In *Proceedings of the Second International Conference on Scale-Space Theories in Computer Vision*, SCALE-SPACE '99, pages 187–198. Springer-Verlag, 1999.

[165] P.A.V. Miranda and L.A.C. Mansilla. Oriented image foresting transform segmentation by seed competition. *IEEE Transactions on Image Processing*, 23:389–398, 2014.

[166] D. Modolo and V. Ferrari. Learning semantic part-based models from google images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1502–1509, 2017.

[167] P. Monasse and F. Guichard. Fast computation of a contrast-invariant image representation. *IEEE Transactions on Image Processing*, 9(5):860–872, 2000.

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

[169] N. Monath, A. Kobren, A. Krishnamurthy, and A. McCallum. Gradient-based hierarchical clustering. In *Advances in Neural Information Processing Systems*, 2017.

[170] E. Monfrini, J. Lecomte, F. Desbouvries, and W. Pieczynski. Image and signal restoration using pairwise markov trees. In *IEEE Workshop on Statistical Signal Processing*, pages 174–177, 2003.

[171] J.M. Morel and S. Solimini. *Variational Methods in Image Segmentation*. Birkhauser Boston Inc., 1995.

[172] U. Moschini, A. Meijster, and M.H.F. Wilkinson. A hybrid shared-memory parallel max-tree algorithm for extreme dynamic-range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):513–526, 2017.

[173] U. Moschini and M.H.F. Wilkinson. Viscous-hyperconnected attribute filters: A first algorithm. In *Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 669–680. Springer International Publishing, 2015.

[174] B. Moseley and J. Wang. Approximation bounds for hierarchical clustering:

Average linkage, bisecting k-means, and local search. In *Advances in Neural Information Processing Systems*, pages 3094–3103, 2017.

[175] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[176] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.

[177] F. Murtagh and P. Contreras. Algorithms for hierarchical clustering: an overview. *Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

[178] B. Naegel and N. Passat. Component-trees and multi-value images: A comparative study. In *Mathematical Morphology and Its Application to Signal and Image Processing*, volume 5720 of *Lecture Notes in Computer Science*, pages 261–271. Springer, 2009.

[179] B. Naegel and N. Passat. Towards connected filtering based on component-graphs. In *Mathematical Morphology and Its Applications to Signal and Image Processing*, volume 7883 of *Lecture Notes in Computer Science*, pages 353–364. Springer, 2013.

[180] B. Naegel and N. Passat. Colour image filtering with component-graphs. In *2014 22nd International Conference on Pattern Recognition*, pages 1621–1626. IEEE, 2014.

[181] B. Naegel, N. Passat, N. Boch, and M. Kocher. Segmentation using vector-attribute filters: methodology and application to dermatological imaging. In *International Symposium on Mathematical Morphology*, volume 1, pages 239–250, 2007.

[182] B. Naegel and L. Wendling. A document binarization method based on connected operators. *Pattern Recognition Letters*, 31(11):1251–1259, 2010.

[183] M. Nagao, T. Matsuyama, and Y. Ikeda. Region extraction and shape analysis in aerial photographs. *Computer Graphics and Image Processing*, 10(3):195–223, 1979.

[184] L. Najman. On the equivalence between hierarchical segmentations and ultrametric watersheds. *Journal of Mathematical Imaging and Vision*, 40(3):231–247, 2011.

[185] L. Najman and M. Couprie. Building the component tree in quasi-linear time. *IEEE Transactions on Image Processing*, 15(11):3531–3539, 2006.

[186] L. Najman and Th. Géraud. Discrete set-valued continuity and interpolation. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 37–48. Springer, 2013.

[187] L. Najman and M. Schmitt. Watershed of a continuous function. *Signal Processing*, 38(1):99–112, 1994.

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

[189] O. Nempont, J. Atif, E. Angelini, and I. Bloch. A new fuzzy connectivity measure for fuzzy sets. *Journal of Mathematical Imaging and Vision*, 34(2):107–136, 2009.

[190] A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856, 2002.

[191] D. Nogly and M. Schladt. Digital topology on graphs. *Computer Vision and Image Understanding*, 63:394—396, 1996.

[192] G.K. Ouzounis, Martino Pesaresi, and P. Soille. Differential area profiles: Decomposition properties and efficient computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1533–1548, 2012.

[193] G.K. Ouzounis and M.H.F. Wilkinson. *A parallel implementation of the dual-input Max-Tree algorithm for attribute filtering*. University of Groningen, Johann Bernoulli Institute for Mathematics, 2007.

[194] G.K. Ouzounis and M.H.F. Wilkinson. Partition-induced connections and operators for pattern analysis. *Pattern Recognition*, 43(10):3193–3207, 2010.

[195] G.K. Ouzounis and M.H.F. Wilkinson. Hyperconnected attribute filters based on k-flat zones. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):224–239, 2011.

[196] N. Passat, B. Naegel, and C. Kurtz. Component-graph construction. *Journal of Mathematical Imaging and Vision*, 61(6):798–823, 2019.

[197] J. Pawley. *Handbook of biological confocal microscopy*, volume 236. Springer Science & Business Media, 2006.

[198] M. Peikari, S. Salama, S. Nofech-Mozes, and A.L. Martel. A cluster-then-label semi-supervised learning approach for pathology image classification. *Scientific reports*, 8(1):1–13, 2018.

[199] W. Pieczynski. Pairwise markov chains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):634–639, 2003.

[200] W. Pieczynski. Triplet markov chains and image segmentation. In *Inverse problems in Vision and 3D Tomography*, pages 123–153. Wiley, 2013.

[201] J. Pont-Tuset, P. Arbelaez, 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, 2016.

[202] J. Pont-Tuset and F. Marques. Supervised assessment of segmentation hierarchies. In *European Conference on Computer Vision*, 2012.

[203] J. Pont-Tuset and F. Marques. Upper-bound assessment of the spatial accuracy

of hierarchical region-based image representations. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2012.

[204] J. Pont-Tuset and F. Marques. Supervised evaluation of image segmentation and object proposal techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1465–1478, 2016.

[205] A. Pothen, H.D. Simon, and K.-P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM journal on matrix analysis and applications*, 11(3):430–452, 1990.

[206] I. Pratikakis, B. Gatos, and K. Ntirogianni. H–dibco 2010 - handwritten document image binarization competition. In *International Conference on Frontiers in Handwriting Recognition*, 2010.

[207] J.-N. Provost, Ch. Collet, Ph. Rostaing, P. Pérez, and P. Bouthemy. Hierarchical markovian segmentation of multispectral images for the reconstruction of water depth maps. *Computer Vision and Image Understanding*, 93(2):155–174, 2004.

[208] Z. Ren and G. Shakhnarovich. Image segmentation by cascaded region agglomeration. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2011–2018, 2013.

[209] G.E. Reyes and H. Zolfaghari. Bi-heyting algebras, toposes and modalities. *Journal of Philosophical Logic*, 25:25–43, 1996.

[210] C. Ronse. Partial partitions, partial connections and connective segmentation. *Journal of Mathematical Imaging and Vision*, 32(2):97–125, 2008.

[211] C. Ronse. Idempotent block splitting on partial partitions, I: Isotone operators. *Order*, 28(2):273–306, 2011.

[212] C. Ronse. Idempotent block splitting on partial partitions, II: Non-isotone operators. *Order*, 28(2):307–339, 2011.

[213] C. Ronse. Orders on partial partitions and maximal partitioning of sets. In *Mathematical Morphology and Its Applications to Image and Signal Processing*, volume 6671 of *Lecture Notes in Computer Science*, pages 49–60. Springer, 2011.

[214] C. Ronse. Axiomatics for oriented connectivity. *Pattern Recognition Letters*, 47:120–128, 2014.

[215] C. Ronse and J. Serra. Geodesy and connectivity in lattices. *Fundamenta Informaticae*, 46(4):349–395, 2001.

[216] A. Rosenfeld. Connectivity in digital pictures. *Journal of the ACM (JACM)*, 17:146–160, 1970.

[217] A. Rosenfeld. Adjacency in digital pictures. *Information and Control*, 26:24–33, 1974.

[218] A. Roy and S. Pokutta. Hierarchical clustering via spreading metrics. In *Advances in Neural Information Processing Systems*, pages 2316–2324. 2016.

[219] A. Roy and S. Pokutta. Hierarchical clustering via spreading metrics. *The*

Journal of Machine Learning Research, 18(1):3077–3111, 2017.

[220] P. Salembier. Connected operators based on tree pruning strategies. In *Mathematical Morphology*, pages 177–198. John Wiley & Sons, Inc., 2013.

[221] P. Salembier and S. Foucher. Optimum graph cuts for pruning binary partition trees of polarimetric sar images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(9):5493–5502, 2016.

[222] 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.

[223] P. Salembier, S. Liesegang, and C. López-Martínez. Ship detection in sar images based on maxtree representation and graph signal processing. *IEEE Transactions on Geoscience and Remote Sensing*, 57(5):2709–2724, 2019.

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

[225] P. Salembier and J. Serra. Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Transactions on Image Processing*, 4(8):1153–1160, 1995.

[226] P. Salembier and M.H.F. Wilkinson. Connected operators: A review of region-based morphological image processing techniques. *IEEE Signal Processing Magazine*, 26:136–157, 2009.

[227] F. Salzenstein and Ch. Collet. Fuzzy markov random fields versus chains for multispectral image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1753–1767, 2006.

[228] J. Serra. Mathematical morphology for Boolean lattices. In *Image Analysis and Mathematical Morphology. II: Theoretical Advances*, pages 37–58. Academic Press, 1988.

[229] J. Serra. Connectivity on complete lattices. *Journal of Mathematical Imaging and Vision*, 9(3):231–251, 1998.

[230] J. Serra. Connections for sets and functions. *Fundamenta Informaticae*, 41(1-2):147–186, 2000.

[231] J. Serra. Morphological segmentations of colour images. In *Mathematical Morphology: 40 Years On*, volume 30 of *Computational Imaging and Vision*, pages 151–176. Springer, 2005.

[232] J. Serra and P. Salembier. Connected operators and pyramids. In *Algebra and Morphological Image Processing IV*, volume 2030 of *SPIE Proceedings*, pages 65–76, 1993.

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

[234] D. Singaraju, L. Grady, and R. Vidal. Interactive image segmentation via

minimization of quadratic energies on directed graphs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[235] P.H.A. Sneath. The application of computers to taxonomy. *Microbiology*, 17(1):201–226, 1957.

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

[237] J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken. Ridge-based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, 23:501–509, 2004.

[238] J.G. Stell. Relations in mathematical morphology with applications to graphs and rough sets. In *COSIT: Spatial Information Theory*, Lecture Notes in Computer Science, pages 438–454, 2007.

[239] J.G. Stell and M.F. Worboys. The algebraic structure of sets of regions. In *COSIT, Spatial Information Theory: A Theoretical Basis for GIS*, Lecture Notes in Computer Science, pages 163–174, 1997.

[240] M. Stoer and F. Wagner. A simple min-cut algorithm. *Journal of the ACM (JACM)*, 44(4):585–591, 1997.

[241] S. Tanimoto and T. Pavlidis. A hierarchical data structure for picture processing. *Computer Graphics and Image Processing*, 4(2):104–119, 1975.

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

[243] P. Teeninga, U. Moschini, S.C. Trager, and M.H.F. Wilkinson. Improved detection of faint extended astronomical objects through statistical attribute filtering. In *Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 157–168. Springer International Publishing, 2015.

[244] P. Teeninga, U. Moschini, S.C. Trager, and M.H.F. Wilkinson. Statistical attribute filtering to detect faint extended astronomical sources. *Mathematical Morphology-Theory and Applications*, 1(1), 2016.

[245] F.B. Tushabe. *Extending Attribute Filters to Color Processing and Multi-Media Applications*. PhD thesis, University of Groningen, 2010.

[246] E.R. Urbach, J.B.T.M. Roerdink, and M.H.F. Wilkinson. Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):272–285, 2007.

[247] S. Vickers. *Topology via Logic*. Cambridge University Press, 1989.

[248] A. Vierkötter, U. Ranft, U. Krämer, D. Sugiri, V. Reimann, and J. Krutmann. The SCINEXA: a novel, validated score to simultaneously assess and differentiate between intrinsic and extrinsic skin ageing. *Journal of dermatological science*, 53(3):207–211, 2009.

[249] S. Vikram and S. Dasgupta. Interactive bayesian hierarchical clustering. In *International Conference on Machine Learning*, pages 2081–2090, 2016.

[250] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):583–598, 1991.

[251] S. Wang and J.M. Siskind. Image segmentation with minimum mean cut. In *IEEE International Conference on Computer Vision*, volume 1, pages 517–524, 2001.

[252] J.H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[253] K.R. Weber and S.T. Acton. On connected operators in color image processing. *Journal of Electronic Imaging*, 13(3):619–629, 2004.

[254] P. Wendt, E.J. Coyle, and Jr. Gallagher, N.C. Stack filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34(4):898–911, 1986.

[255] M.A. Westenberg, J.B.T.M. Roerdink, and M.H.F. Wilkinson. Volumetric attribute filtering and interactive visualization using the max-tree representation. *IEEE Transactions on Image Processing*, 16(12):2943–2952, 2007.

[256] J. Whittaker. *Graphical models in applied multivariate statistics*. Wiley, 1990.

[257] M.H.F. Wilkinson. Attribute-space connectivity and connected filters. *Image Vision Computing*, 25(4):426–435, 2007.

[258] M.H.F. Wilkinson. An axiomatic approach to hyperconnectivity. In *Mathematical Morphology and Its Application to Signal and Image Processing*, volume 5720 of *Lecture Notes in Computer Science*, pages 35–46. Springer, 2009.

[259] M.H.F. Wilkinson. Hyperconnectivity, attribute-space connectivity and path openings: Theoretical relationships. In *Mathematical Morphology and Its Application to Signal and Image Processing*, volume 5720 of *Lecture Notes in Computer Science*, pages 47–58. Springer, 2009.

[260] M.H.F. Wilkinson. A fast component-tree algorithm for high dynamic-range images and second generation connectivity. In *2011 18th IEEE International Conference on Image Processing*, pages 1021–1024, 2011.

[261] M.H.F. Wilkinson. Hyperconnections and openings on complete lattices. In *Mathematical Morphology and Its Application to Signal and Image Processing*, volume 6671 of *Lecture Notes in Computer Science*, pages 73–84. Springer, 2011.

[262] M.H.F. Wilkinson, H. Gao, W.H. Hesselink, J.-E. Jonker, and A. Meijster. Concurrent computation of attribute filters on shared memory parallel machines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1800–1813, 2008.

[263] Y. Wu and J.Y. Qu. Autofluorescence spectroscopy of epithelial tissues. *Journal of biomedical optics*, 11(5):054023, 2006.

[264] E.M.T. Wurm, C. Longo, C. Curchin, H.P. Soyer, T.W. Prow, and G. Pellacani. In vivo assessment of chronological ageing and photoageing in forearm skin using reflectance confocal microscopy. *British Journal of Dermatology*, 167(2):270–279, 2012.

[265] Y. Xu, E. Carlinet, Th. Géraud, and L. Najman. Hierarchical segmentation using tree-based shape spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(3):457–469, 2017.

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

[267] Y. Xu, Th. Géraud, and L. Najman. Hierarchical image simplification and segmentation based on mumford–shah-salient level line selection. *Pattern Recognition Letters*, 83:278 – 286, 2016.

[268] Y. Xu, P. Monasse, Th. Géraud, and L. Najman. Tree-Based Morse Regions: A Topological Approach to Local Feature Detection. *IEEE Transactions on Image Processing*, 23(12):5612–5625, 2014.

[269] N. Yadav, A. Kobren, N. Monath, and A. Mccallum. Supervised hierarchical clustering with exponential linkage. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 6973–6983. PMLR, 2019.

[270] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1155–1162, 2013.

[271] J.E. Yarkony and C. Fowlkes. Planar ultrametrics for image segmentation. In *Advances in Neural Information Processing Systems*, pages 64–72, 2015.

[272] D.G. York, J. Adelman, Jr. Anderson, J.E., S.F. Anderson, J. Annis, N.A. Bahcall, J. A. Bakken, R. Barkhouser, S. Bastian, E. Berman, W.N. Boroski, S. Bracker, C. Briegel, J.W. Briggs, J. Brinkmann, R. Brunner, S. Burles, L. Carey, M.A. Carr, F.J. Castander, B. Chen, P.L. Colestock, A. J. Connolly, J. H. Crocker, I. Csabai, P.C. Czarapata, J.E. Davis, M. Doi, T. Dombeck, D. Eisenstein, N. Ellman, B.R. Elms, M.L. Evans, X. Fan, G.R. Federwitz, L. Fiscelli, S. Friedman, J.A. Frieman, M. Fukugita, B. Gillespie, J.E. Gunn, V.K. Gurbani, E. de Haas, M. Haldeman, F.H. Harris, J. Hayes, T.M. Heckman, G. S. Hennessy, R.B. Hindsley, S. Holm, D.J. Holmgren, C. Huang, C. Hull, D. Husby, S. Ichikawa, T. Ichikawa, Ž. Ivezić, S. Kent, R.S. J. Kim, E. Kinney, M. Klaene, A. N. Kleinman, S. Kleinman, G. R. Knapp, J. Korienek, R.G. Kron, P.Z. Kunszt, D. Q. Lamb, B. Lee, R.F. Leger, S. Limmongkol, C. Lindenmeyer, D.C. Long, C. Loomis, J. Loveday, R. Lucinio, R.H. Lupton, B. MacKinnon, E.J. Mannery, P. M. Mantsch, B. Margon, P. McGehee, T.A. McKay, A. Meiksin, A. Merelli, D.G. Monet, J.A. Munn, V.K. Narayanan, T. Nash, E. Neilsen, R. Neswold, H.J. Newberg, R. C. Nichol, T. Nicinski, M. Nonino, N. Okada, S. Okamura, J.P. Ostriker, R. Owen, A.G. Pauls, J. Peoples, R. L. Peterson, D. Petravick, J.R. Pier, A. Pope, R. Pordes, A. Prosapio, R. Rechenmacher, T.R. Quinn, G.T. Richards, M.W. Richmond, C.H. Rivetta, C.M. Rockosi,

K. Ruthmansdorfer, D. Sand ford, D.J. Schlegel, D.P. Schneider, M. Sekiguchi, G. Sergey, K. Shimasaku, W.A. Siegmund, S. Smee, J.A. Smith, S. Snedden, R. Stone, C. Stoughton, M.A. Strauss, C. Stubbs, M. SubbaRao, A.S. Szalay, I. Szapudi, G.P. Szokoly, A.R. Thakar, C. Tremonti, D.L. Tucker, A. Uomoto, D. Vanden Berk, M.S. Vogeley, P. Waddell, S. Wang, M. Watanabe, D.H. Weinberg, B. Yanny, N. Yasuda, and SDSS Collaboration. The Sloan Digital Sky Survey: Technical Summary. *The Astronomical Journal*, 120(3):1579–1587, 2000.

[273] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern recognition*, 37(1):1–19, 2004.

[274] Q. Zhao. Segmenting natural images with the least effort as humans. In *The British Machine Vision Conference*, pages 110.1–110.12, 2015.

Appendix A

Detailed curriculum vitae

A.1 Resume

Personal information

Name	Benjamin Perret
Birth date	22 October 1984
Nationality	French
Professional address	ESIEE Paris, 2 boulevard Blaise Pascal Cité Descartes BP 99 93162 Noisy le Grand cedex
Phone number	+33 (0)1 45 92 67 09
E-Mail	benjamin.perret@esiee.fr
Web page	`http://www.esiee.fr/~perretb/`

Current position

Since 2011:

- Associate Professor, ESIEE Paris - Université Gustave Eiffel, Computer science department
- Member of the Laboratoire d'Informatique Gaspard Monge (LIGM - UMR 8049), A3SI Team

Professional experience

2018-2019 Scientific visitor: 1 year full time **CNRS delegation**, LAMSADE, Université Paris Dauphine, machine learning team

2010-2011 **Teaching Assistant** (ATER), Université de Strasbourg, LSIIT (UMR 7005), UFR Mathematics and Computer Science
- **Teaching:** 192 HETD (*heures équivalent TD*) of teaching.

2007-2010 **PhD candidate in Computer Science**, Université de Strasbourg, LSIIT (UMR 7005)
- **Subject:** Caractérisation multibande de galaxies par hiérarchie de modèles et arbres de composantes connexes.
- **Directors:** Ch. Collet and E. Slezak
- **Advisors:** S. Lefèvre and V. Mazet
- **Jury:** J. Serra (reviewer, president), X. Descombes (reviewer), E. Bertin, Ch. Collet, E. Slezak, S. Lefèvre, V. Mazet

2007-2010 **Teaching Assistant** (moniteur), Université de Strasbourg, UFR Mathematics and Computer Science
- **Teaching:** 64 HETD (*heures équivalent TD*) of teaching per year.
- **Training:** 30 days of teacher and scientific communication training

2007 **Research internship**, Université de Strasbourg, LSIIT (UMR 7005)
- **Subject:** Détection des galaxies à faible brillance de surface.
- **Directors:** Ch. Collet and S. Lefèvre

2006 **Developer Internship**, Centre de Données astronomiques de Strasbourg, Observatoire Astronomique de Strasbourg
- **Subject:** Online image processing services for the virtual observatory.
- **Directors:** F. Bonnarel and Ch. Collet

Education

2010 **PhD in Computer Science**, Université de Strasbourg, LSIIT (UMR 7005)

2007 **MSc in Fundamental and Applied Computer Science**, Université de Strasbourg, LSIIT (UMR 7005)

Award

- 2010 – Best **thesis award** from Université de Strasbourg.

A.2 Research

Overview

Publications

International journals	15
International conferences	24
National conference	1
Book chapters	2

Student Supervision

Completed PhD thesis	2
Ongoing PhD thesis	2
Internships	9

PhD Students supervision

Gabriel Barbosa da Fonseca 2019-present

Subject:	Supervised segmentation for image and video segmentation
Advisers:	with J. Cousty, R. Negrel, and Silvio J.F. Guimarães
Founding:	CAPES/COFECUB project HIMMD

Thanh Nguyen Xuan 2018-present

Subject:	Faint object detections in multiband astronomical images
Advisers:	with G. Chierchia, L. Najman, and H. Talbot
Founding:	European ITN H2020 SUNDIAL

Deise Santana Maia 2016-2019

Subject:	A study of hierarchical watersheds on graphs with applications to image segmentation
Advisers:	with J. Cousty and L. Najman
Founding:	Labex Bézout and ESIEE Paris

Julie Robic 2015-2018

Subject:	Automated quantification of the skin aging process using in-vivo confocal microscopy
Advisers:	with M. Couprie, A. Nkengne, and H. Talbot
Founding:	CIFRE with Clarins Laboratories

Internships supervision

Carla Yagoub 2020

Subject:	Hyperbolic embeddings of ultrametrics
Advisers:	with G. Chierchia and H. Talbot
Origin:	CentraleSupélec - ENS Cachan Master MVA

Wagner Rodrigues 2020

Subject:	Supervised end-to-end learning of Hierarchical Clustering via Projection to Target Walking
Advisers:	with G. Chierchia and H. Talbot
Origin:	Polytechnique

Pierre Kramer 2020

Subject:	Out-of-core Algorithms for Hierarchical Watershed
Advisers:	with J. Cousty and H. Phelippeau (Thermo Fisher Scientific)
Origin:	ENSEIRB-MATMECA, Bordeaux INP

Stela Carneiro Espíndola 2019

Subject:	Out-of-Core Construction of a Binary Partition Tree for a Watershed Hierarchy
Advisers:	with J. Cousty
Origin:	ESIEE Paris

Salah Boudjadja 2019

Subject:	Creation and detection of deep fakes
Advisers:	with V. Nozick and M. Pic (Surys - Hologram Industries)
Origin:	Université Paris Est, Master *Sciences de l'image*

Karim Sadki 2019

Subject:	Real-time falsification of identity cards
Advisers:	with V. Nozick and M. Pic (Surys - Hologram Industries)
Origin:	Université Paris Est, Master *Sciences de l'image*

Deise Santana Maia 2016

Subject:	Combination of hierarchies of segmentations
Advisers:	with J. Cousty
Origin:	Master Bézout

Mohamad Onayssi 2016

Subject:	Edge weight functions for graph-based hierarchical image analysis
Advisers:	with J. Cousty
Origin:	ESIEE Paris

Jean Carlo Rivera Ura 2014

Subject: Supervised assessment of hierarchical image segmentation
Advisers: with J. Cousty
Origin: ESIEE Paris

Projects and contracts

ANR JCJC ULTRA-LEARN 2021-2025

Subject: Supervised Ultrametric Learning
Position: Coordinator

Industrial contract with Thermo Fisher Scientific 2019-2020

Subject: Out-of-core algorithms for large image segmentation
Position: Member

CAPES/COFECUB HiMMD (Grant 933/19) 2018-2022

Subject: Hierarchical Graph-based Analysis of Image, Video and Multimedia Data
Position: Member
Consortium: Pontifical Catholic University of Minas Gerais (PUC Minas, Brazil), Laboratoire d'Informatique Gaspard Monge (LIGM), Universidade Federal de Minas Gerais (UFMG, Brazil), University of Campinas (UNICAMP, Brazil), Grenoble Institute of Technology (Grenoble INP), Institut de recherche en informatique et systèmes aléatoires (IRISA)

H2020 ITN SUNDIAL (Grant 721463) 2017-2021

Subject: SUrvey Network for Deep Imaging Analysis and Learning
Position: Member
Consortium: University of Groningen, Univeristy of Oulu, University of Birmingham, University of Ghent, ESIEE Paris, Instituto de Astrofísica de Canarias, University of Naples Federico II, Osservatorio di Capodimonte, University of Heidelberg
Website: `https://www.astro.rug.nl/~sundial/`

Industrial contract with Clarins Laboratories 2015-2018

Subject: Automatic characterization of skin aging in confocal microscopy
Position: Coordinator

CAPES/COFECUB (Grant 933/19) 2008-2014

Subject:	Hierarchical Graph-based Image and Video Segmentation
Position:	Member
Consortium:	Universidade Federal de Minas Gerais (UFMG, Brazil), University of Campinas (UNICAMP, Brazil), ENSEA, ESIEE Paris, Université Paris 6

ANR DAHLIA (08-BLAN-0253-01) 2008-2012

Subject:	Dedicated Algorithms for HyperspectraL Imaging in Astronomy
Position:	Member
Consortium:	Observatoire de la Cote d'Azur (OCA), Laboratoire d'Astrophysique de Toulouse (LATT), Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection (LSIIT), Centre de Recherche Astrophysique de Lyon (CRAL)
Website:	`https://dahlia.oca.eu/`

Software and data produced

- Development of the Python/C++ library for **hierarchical graph analysis HIGRA** `https://github.com/higra/Higra`
- Development of a Python **optimization toolbox for hierarchical clustering** `https://github.com/PerretB/ultrametric-fitting`
- Development of a Python library to design and compute **directed connected filters** `https://perso.esiee.fr/~perretb/dc-hierarchy.html`
- Development of an online demonstrator for **marker based hierarchical segmentation** `https://perso.esiee.fr/~perretb/ISeg/`
- Development of a c++ **assessment framework for hierarchies of segmentations** and the associated ground-truth dataset `https://perso.esiee.fr/~perretb/supeval.html`

Editorial work and scientific animation

- Reviewer for various journals and conferences: IEEE Transactions on Image Processing, Pattern Recognition, Journal of Mathematical Imaging and Vision, Pattern Recognition Letters, Journal of Signal Processing: Image Communication, Journal of Fuzzy Sets and Systems, Journal of Mathematical Morphology-Theory and Applications, Journal of Optics & Laser Technology, Journal of Computers & Electrical Engineering, Journal of Applied Mathematical Modelling, Journal of Applied Mathematics and Computation, Journal of Computer Methods and Programs in Biomedicine, International Symposium on Mathematical Morphology, International Conference on Discrete Geome-

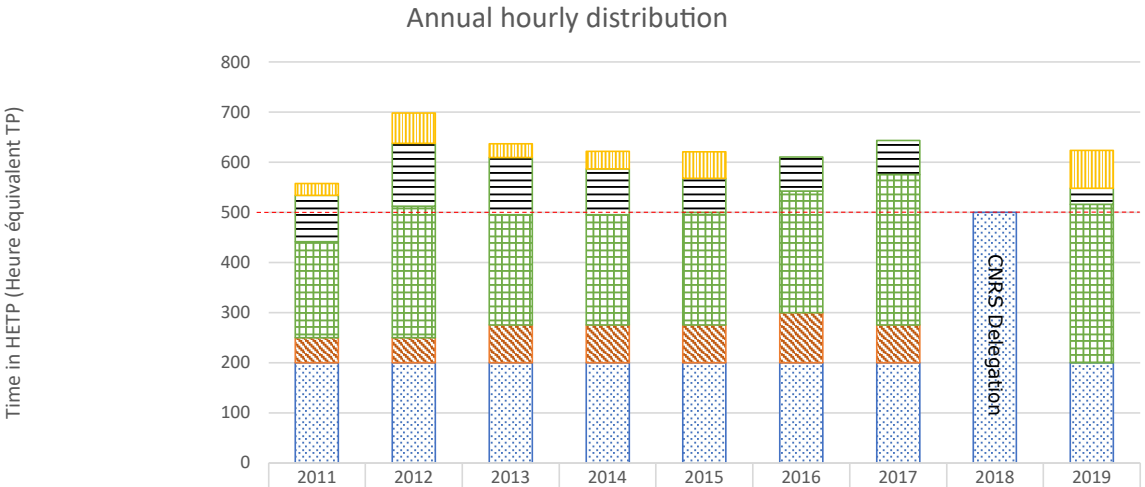
try for Computer Imagery, International Symposium on Biomedical Imaging, International Conference on Pattern Recognition and Artificial Intelligence

- Member of the program comity of ISMM 2017 (International Symposium on Mathematical Morphology).
- Member of the organizing comity of ISMM 2019 (International Symposium on Mathematical Morphology)
- Guest editor for a special issue in the Journal of Mathematical Morphology-Theory and Applications following ISMM 2019.

A.3 Teaching

Overview

The following chart shows my annual hourly distribution since my recruitment at ESIEE Paris in 2011. The unit of measurement is the HETP (*Heure Équivalent Travaux Pratique*) which is equivalent to 2/3 of a HETD (*Heure Équivalent Travaux Dirigé*) used in French Universities. A full time annual service is composed of 500 HETP. Teacher-researchers have a 200 HETP discharge for their research activities, the remaining 300 HETP are dedicated to teaching, student tutoring, student project supervision, and administrative tasks related to pedagogical activities.



	2011	2012	2013	2014	2015	2016	2017	2018	2019
Student project supervision	24	60	27	35	53	0	0	0	76
Apprenticeship tutoring	92	126	115	92	68	68	68	0	32
Teaching	192	262	220	220	225	243	301	0	316
Administrative responsability	50	50	75	75	75	100	75	0	0
Research	200	200	200	200	200	200	200	500	200

Principal responsibilities

2011-2018 **Coordinator of a 3-year engineering degree** in apprenticeship in computer science specialized in computer vision and virtual reality, ESIEE Paris.

- definition of educational programs;
- animation of the teaching team;
- promotion of the degree in undergraduate schools;
- selection of candidate students;
- relations with enterprises; and
- management of a 60k € yearly funding (fitting out of premises and purchasing of equipment).

Main courses

Image processing and analysis since 2013

Abstract:	1) sensors, sampling and quantization, 2) intensity transformation and histograms, 3) adjacency, connected components, and geometric transformations, 4) linear spatial filters, 5) non-linear spatial filters, 6) introduction to research
Volume:	30 hours
Public:	4th year
Position:	Course creator and teacher
Webpage:	`https://perso.esiee.fr/~perretb/I5FM/TAI/`

Object oriented programming since 2012

Abstract:	1) Introduction to C♯, 2) Classes and objects, 3) Inheritance, 4) Project development: reproduction of the game Space Invaders
Volume:	30 hours
Public:	3rd year
Position:	Course creator and teacher
Webpage:	`https://perso.esiee.fr/~perretb/I3FM/POO1/`

Linear algebra and 3D programming since 2012

Abstract:	Development from scratch of a simple wire-frame 3D engine 1) vectors , 2) linear applications and matrices, 3) composition of linear applications and matrix product, 4) homogeneous coordinates, geometric transformations, and projections.
Volume:	12 hours
Public:	3rd year
Position:	Course creator and teacher
Webpage:	`https://perso.esiee.fr/~perretb/a3pal/`

Mathematical morphology since 2015

Abstract:	Connected filters and connected image filtering. 1) binary and grey-scale opening and closing by reconstruction, 2) binary and grey-scale opening and closing connected filters, 3) component trees, 4) attribute connected filters
Volume:	6 hours (total course volume: 30 hours)
Public:	5th year
Position:	Course co-creator and teacher
Webpage:	`https://perso.esiee.fr/~perretb/MM/`

Optimization since 2019

Abstract:	1) Array programming, 2) zero order optimization, 3) gradient descent optimization, 4) neural networks, 5) hyper-parameter optimization, 6) optimization problem modeling and solving
Volume:	30 hours
Public:	5th year
Position:	Teacher

Introduction to artificial intelligence in games since 2019

Abstract:	1) Monte Carlo AI, illustration on the game *Tron*, 2) Min-Max algorithms, illustration on the game *Tic-Tac-Toe*, 3) Shortest-path algorithms, illustration on the game *Pac Man*, 4) exploratory project: alpha-beta algorithm, Q-learning, reinforcement learning.
Volume:	40 hours
Public:	2th year
Position:	Teacher