



A short tour of mathematical morphology on edge and vertex weighted graphs

Laurent Najman, Fernand Meyer

► To cite this version:

Laurent Najman, Fernand Meyer. A short tour of mathematical morphology on edge and vertex weighted graphs. Lezoray, O. and Grady, L. Image Processing and Analysis with Graphs: Theory and Practice, CRC Press, pp.141-174, 2012, Digital Imaging and Computer Vision, 9781439855072. hal-00741976

HAL Id: hal-00741976

<https://hal.science/hal-00741976>

Submitted on 15 Oct 2012

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

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

Chapter 1

A short tour of mathematical morphology on edge and vertex weighted graphs

LAURENT NAJMAN #1

Université Paris-Est

Laboratoire d'Informatique Gaspard-Monge - Equipe A3SI - ESIEE Paris
France

Email: l.najman@esiee.fr

FERNAND MEYER #2

Centre de Morphologie Mathématique - Ecole des Mines de Paris
Fontainebleau, France

Email: f.meyer@cmm.ensmp.fr

1.1 Introduction

Mathematical morphology is a discipline of image analysis that was introduced in the mid-1960s by two researchers at the École des Mines in Paris: Georges Matheron [1] and Jean Serra [2, 3]. Historically, it was the first consistent non-linear image analysis theory, which from the very start included not only theoretical results but also many practical aspects. Due to the algebraic nature of morphology, the space on which the operators are defined can be either continuous or discrete. However, it was only in 1989 [4] that researchers from the CMM at the École des Mines began to study morphology on graphs, soon formalized in

[5]. Fairly recent development [6, 7, 8, 9, 10, 11, 12] in this direction has several motivations and beneficial consequences that we are going to review in this chapter. Rather than trying to cover every aspect of the theory, we choose to present a comprehensive subset based on a recent unifying graph theoretical framework developed by the A3SI team of the LIGM at Paris-Est University. The presentation is roughly divided in three parts, dealing respectively with basic operators (mainly based on [10]), hierarchical segmentation (mainly based on [13] and [14]), and optimization (mainly based on [15]). For the reader interested in a more complete presentation of morphology, we recommend [16] and the recent [17].

One of the fundamental ideas of mathematical morphology is to compare unknown objects with known ones. We begin by presenting the tools that make such an idea practicable: mathematical structures called *lattices* (Section 1.2), allowing us to compare weighted and non-weighted edges and vertices. We then present (Section 1.3) several dilations and erosions, that always come in pairs (they are called adjunct operators). From there, we can build (Section 1.4) some morphological filters, called opening or closing. During the presentation of those various operators, we often give an interpretation of classical graph-operators in morphological terms. We conclude this part by presenting (Section 1.5) some connected operators based on pruning a tree representation of the image, illustrating their usage for image filtering and simplification.

The second part of the chapter deals with hierarchical segmentation. During the course of the chapter's first part, several times the minimum spanning tree appears. This tree is the oldest combinatorial optimization problem [18, 19]. Since the seminal work of Zahn [20], the minimum spanning tree has been extensively used in classification. Its first appearance in image processing dates from 1986, in a paper by Morris *et al.* [21]. Meyer was the first to explicitly use it in a morphological context [22]. A strong momentum to its usage in segmentation has been provided thanks to a 2004 paper by Felzenswalb and Huttenlocher [23]. In the second part of the chapter, we revisit the watershed [24], and show that in the framework of edge-weighted graphs, the watershed has very strong links with the minimum spanning tree (Section 1.6). Thanks to such links, we can use the watershed for building hierarchies of segmentations (Section 1.7), relying on the filtering tools seen in the first part of the chapter.

The main principle of morphology, comparison, is rather different from the optimization paradigm. However, rather than opposing these two viewpoints, it is more fruitful to explore their connections. In the last part of the chapter (Section 1.8), we turn towards optimization and show that the watershed presented in the previous part can be extended to be used as an optimization tool.

1.2 Graphs and lattices

In mathematical morphology, we compare objects one with respect to each other. The mathematical structure that allows us to make such an operation effective is called a lattice. Recall that a (complete) *lattice* is a

partially ordered set, that also has a least upper bound, called *supremum*, and a greatest lower bound, called *infimum*. More formally, a *lattice* [25] (\mathcal{L}, \leq) is a set \mathcal{L} (the space) endowed with an *ordering* relationship \leq , which is reflexive ($\forall x \in \mathcal{L}, x \leq x$), anti-symmetric ($x \leq y$ and $y \leq x \implies x = y$) and transitive ($x \leq y$ and $y \leq z \implies x \leq z$). This ordering is such that for all x and y , we can define both a larger element $x \vee y$ and a smaller element $x \wedge y$. Such a lattice is said to be *complete* if any subset \mathcal{P} of \mathcal{L} has a *supremum* $\bigvee \mathcal{P}$ and an *infimum* $\bigwedge \mathcal{P}$ that both belongs to \mathcal{L} . The supremum is formally the smallest amongst all elements of \mathcal{L} that are greater than all the elements of \mathcal{P} , and conversely the infimum is the largest element of \mathcal{L} that is smaller than all the elements of \mathcal{P} .

Most of the morphological theory can be presented and developed at this abstract level, without making references to the properties of the underlying space. However, studying what impact such properties have can indeed be interesting in some situations. In the sequel, we study some lattices that can be built from graph spaces, and what kind of (morphological) operators can be built from such lattices.

1.2.1 Lattice of graphs

We define a *graph* as a pair $\mathcal{G} = (\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ where $\mathcal{V}(\mathcal{G})$ is a set and $\mathcal{E}(\mathcal{G})$ is composed of unordered pairs of distinct elements in $\mathcal{V}(\mathcal{G})$, *i.e.*, $\mathcal{E}(\mathcal{G})$ is a subset of $\{\{v_1, v_2\} \subseteq \mathcal{V}(\mathcal{G}) \mid v_1 \neq v_2\}$. Each element of $\mathcal{V}(\mathcal{G})$ is called a *vertex or a point (of \mathcal{G})*, and each element of $\mathcal{E}(\mathcal{G})$ is called an *edge (of \mathcal{G})*. In the sequel, to simplify the notations, e_{ij} stands for the edge $\{v_i, v_j\} \in \mathcal{E}(\mathcal{G})$.

Let \mathcal{G}_1 and \mathcal{G}_2 be two graphs. If $\mathcal{V}(\mathcal{G}_2) \subseteq \mathcal{V}(\mathcal{G}_1)$ and $\mathcal{E}(\mathcal{G}_2) \subseteq \mathcal{E}(\mathcal{G}_1)$, then \mathcal{G}_1 and \mathcal{G}_2 are ordered and we write $\mathcal{G}_2 \sqsubseteq \mathcal{G}_1$. If $\mathcal{G}_2 \sqsubseteq \mathcal{G}_1$, we say that \mathcal{G}_2 is a *subgraph* of \mathcal{G}_1 , or that \mathcal{G}_2 is *smaller* than \mathcal{G}_1 and that \mathcal{G}_1 is *greater* than \mathcal{G}_2 .

Important remark. Hereafter, the workspace is a graph $\mathcal{G} = (\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ and we consider the sets $\mathcal{V}(\mathbb{G})$, $\mathcal{E}(\mathbb{G})$ and \mathbb{G} of respectively all subsets of $\mathcal{V}(\mathcal{G})$, all subsets of $\mathcal{E}(\mathcal{G})$ and all subgraphs of \mathcal{G} . We also use the classical notations $\mathcal{V} = \mathcal{V}(\mathcal{G})$ and $\mathcal{E} = \mathcal{E}(\mathcal{G})$.

Let $\mathcal{S}_0, \mathcal{S}_1 \subseteq \mathbb{G}$ be the sets of respectively the graphs made of a single vertex and the graphs made of a pair of vertices linked by an edge, *i.e.*, $\mathcal{S}_0 = \{\{\{v\}, \emptyset\} \mid v \in \mathcal{V}(\mathcal{G})\}$ and $\mathcal{S}_1 = \{\{\{v_i, v_j\}, \{e_{ij}\}\} \mid e_{ij} \in \mathcal{E}(\mathcal{G})\}$. We set $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1$. Any graph $\mathcal{G}^1 \in \mathbb{G}$ is *generated* by the family $\mathcal{F} = \{\mathcal{G}_1, \dots, \mathcal{G}_\ell\}$ of all elements in \mathcal{S} smaller than \mathcal{G}^1 : $\mathcal{G}^1 = (\bigcup_{i \in [1, \ell]} \mathcal{V}(\mathcal{G}_i), \bigcup_{i \in [1, \ell]} \mathcal{E}(\mathcal{G}_i))$; we say that the elements of \mathcal{F} are the *generators* of \mathcal{G}^1 [10]. Conversely, any family \mathcal{F} of elements in \mathcal{S} generates an element of \mathbb{G} . Hence, \mathcal{S} (*sup-*) *generates* \mathbb{G} .

Clearly, the ordering \sqsubseteq on graphs amount to say that $\mathcal{G}_2 \sqsubseteq \mathcal{G}_1$ when all generators of \mathcal{G}_2 are also generators of \mathcal{G}_1 . Therefore, ordering \sqsubseteq provides a *lattice* structure on the set \mathbb{G} . Indeed, the largest graph smaller than a family $\mathcal{F} = \{\mathcal{G}_1, \dots, \mathcal{G}_\ell\}$ of elements in \mathbb{G} is the graph generated by the generators common to all \mathcal{G}_i , $i \in [1, \ell]$; this *infimum* is denoted by $\bigcap \mathcal{F}$. Similarly, the *supremum* $\bigcup \mathcal{F}$ is generated by the union

of the families of generators of all \mathcal{G}_i , $i \in [1, \ell]$.

If $\mathcal{V}(\mathcal{G}_1) \subseteq \mathcal{V}(\mathcal{G})$ (resp. $\mathcal{E}(\mathcal{G}_2) \subseteq \mathcal{E}(\mathcal{G})$), we denote by $\overline{\mathcal{V}(\mathcal{G}_1)}$ (resp. $\overline{\mathcal{E}(\mathcal{G}_2)}$) the *complementary set of $\mathcal{V}(\mathcal{G}_1)$ (resp. $\mathcal{E}(\mathcal{G}_2)$) in $\mathcal{V}(\mathcal{G})$ (resp. $\mathcal{E}(\mathcal{G})$)*, that is $\overline{\mathcal{V}(\mathcal{G}_1)} = \mathcal{V}(\mathcal{G}) \setminus \mathcal{V}(\mathcal{G}_1)$ (resp. $\overline{\mathcal{E}(\mathcal{G}_2)} = \mathcal{E}(\mathcal{G}) \setminus \mathcal{E}(\mathcal{G}_2)$). Observe that, if \mathcal{G}_1 is a subgraph of \mathcal{G} , then, except in some degenerated cases, the pair $(\overline{\mathcal{V}(\mathcal{G}_1)}, \overline{\mathcal{E}(\mathcal{G}_1)})$ is not a graph.

Property 1 ([10])

The set \mathbb{G} of the subgraphs of \mathcal{G} form a complete lattice, sup-generated by the set $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1$, but not complemented. The supremum and the infimum of any family $\mathcal{F} = \{\mathcal{G}_1, \dots, \mathcal{G}_\ell\}$ of elements in \mathbb{G} are given by respectively $\sqcap \mathcal{F} = (\bigcap_{i \in [1, \ell]} \mathcal{V}(\mathcal{G}_i), \bigcap_{i \in [1, \ell]} \mathcal{E}(\mathcal{G}_i))$ and $\sqcup \mathcal{F} = (\bigcup_{i \in [1, \ell]} \mathcal{V}(\mathcal{G}_i), \bigcup_{i \in [1, \ell]} \mathcal{E}(\mathcal{G}_i))$.

1.2.2 Lattice of weights

As a fixed grid is able to represent images by assigning grey tones to pixels, the graph \mathcal{G} is able to generate a number of derived graphs by assigning weights to the vertices and edges of the graph \mathcal{G} . According to the applications, the weights can be real or integer, taking their values in $\mathbb{R}, \mathbb{R}^+, \mathbb{N}, [-n, +n], [0, +n]$. The weight of a vertice v_i is written w_i , while the one of an edge e_{ij} is written w_{ij} . The set of all (edges and vertices) weights is written w . The case where the weights are binary, belonging to $\{0, 1\}$ may be interpreted as presence/absence: $w_{ij} = 1$ and $w_k = 1$ express respectively the existence of an edge e_{ij} and of a vertice v_k . All edges and vertices of \mathcal{G} with weight 0 do not exist in this weighted graph.

A possible lattice structure on the weights is given by the following. Let w^1 and w^2 be two sets of weights, we have $w^1 \prec w^2$ whenever $w_{ij}^1 < w_{ij}^2$ and $w_k^1 < w_k^2$. The supremum (resp. infimum) of a family of sets of weights is a set of weights where the weight of a given element is the greatest (resp. lowest) possible weight of the weights of the same element in the family.

Remark 2

In a graph, there may be isolated vertices, that is vertices which are not adjacent to an edge. On the contrary, each edge is adjacent to two vertices. For this reason not any binary distribution of weights on the vertices and edges of \mathcal{G} represents a graph. It is the case if and only if each edge with weight 1 is adjacent to two vertices with weight 1. The same holds for any weight distributions. w correspond to a graph if each edge with weight w_{ij} is such that $w_{ij} \leq w_i$ and $w_{ij} \leq w_j$. In other words, this is the case if the extremities of an edge have weights which are not lower than the weight of the edge. Graphs may be used for modelling many different structures. In some cases vertex weight only have a physical meaning and the edge weights simply serve for storing intermediate results in the computation of the weights of the vertices. In other cases it is the converse. In such situations one does not care whether the weights represent a graph. In other situations on the contrary, one cares to define operators transforming an graph into another graph.

1.3 Neighborhood operations on graphs

Morphology really starts if one considers the neighborhood relations between vertices and edges. We now define operators taking as arguments such neighborhoods. The construction is incremental, from the smallest neighborhood to larger neighborhoods.

1.3.1 Adjunctions on graphs

In the graph \mathcal{G} , we can consider sets of points as well as sets of edges. Therefore, it is convenient to consider operators that go from one kind of sets to the other one. In this section, we investigate such operators and we study their morphological properties. Then, based on these operators, we propose several dilations and erosions acting on the lattice of all subgraphs of \mathcal{G} .

Let $\mathcal{V}(\mathcal{G}_1)$ be a subset of $\mathcal{V}(\mathcal{G})$, we denote by $\mathbb{G}_{\mathcal{V}(\mathcal{G}_1)}$ the set of all subgraphs of \mathcal{G} whose vertex set is $\mathcal{V}(\mathcal{G}_1)$. Let $\mathcal{E}(\mathcal{G}_2)$ be a subset of $\mathcal{E}(\mathcal{G})$. We denote by $\mathbb{G}_{\mathcal{E}(\mathcal{G}_2)}$ the set of all subgraphs of \mathcal{G} whose edge set is $\mathcal{E}(\mathcal{G}_2)$.

Definition 3 (edge-vertex correspondences, [10])

We define the operators $\delta_{\mathcal{E}\mathcal{V}}, \epsilon_{\mathcal{E}\mathcal{V}}$ from $\mathcal{E}(\mathbb{G})$ into $\mathcal{V}(\mathbb{G})$ and the operators $\epsilon_{\mathcal{V}\mathcal{E}}, \delta_{\mathcal{V}\mathcal{E}}$ from $\mathcal{V}(\mathbb{G})$ into $\mathcal{E}(\mathbb{G})$ as follows:

	$\mathcal{E}(\mathbb{G}) \rightarrow \mathcal{V}(\mathbb{G})$	$\mathcal{V}(\mathbb{G}) \rightarrow \mathcal{E}(\mathbb{G})$
Provide the object with a graph structure	$\mathcal{E}(\mathcal{G}_1) \rightarrow \delta_{\mathcal{E}\mathcal{V}}(\mathcal{E}(\mathcal{G}_1))$ such that $(\delta_{\mathcal{E}\mathcal{V}}(\mathcal{E}(\mathcal{G}_1)), \mathcal{E}(\mathcal{G}_1)) = \sqcap \mathbb{G}_{\mathcal{E}(\mathcal{G}_1)}$	$\mathcal{V}(\mathcal{G}_1) \rightarrow \epsilon_{\mathcal{V}\mathcal{E}}(\mathcal{V}(\mathcal{G}_1))$ such that $(\mathcal{V}(\mathcal{G}_1), \epsilon_{\mathcal{V}\mathcal{E}}(\mathcal{V}(\mathcal{G}_1))) = \sqcup \mathbb{G}_{\mathcal{V}(\mathcal{G}_1)}$
Provide its complement with a graph structure	$\mathcal{E}(\mathcal{G}_1) \rightarrow \epsilon_{\mathcal{E}\mathcal{V}}(\mathcal{E}(\mathcal{G}_1))$ such that $(\overline{\epsilon_{\mathcal{E}\mathcal{V}}(\mathcal{E}(\mathcal{G}_1))}, \overline{\mathcal{E}(\mathcal{G}_1)}) = \sqcap \mathbb{G}_{\overline{\mathcal{E}(\mathcal{G}_1)}}$	$\mathcal{V}(\mathcal{G}_1) \rightarrow \delta_{\mathcal{V}\mathcal{E}}(\mathcal{V}(\mathcal{G}_1))$ such that $(\overline{\mathcal{V}(\mathcal{G}_1)}, \overline{\delta_{\mathcal{V}\mathcal{E}}(\mathcal{V}(\mathcal{G}_1))}) = \sqcup \mathbb{G}_{\overline{\mathcal{V}(\mathcal{G}_1)}}$

In other words, if $\mathcal{V}(\mathcal{G}_1) \subseteq \mathcal{V}(\mathcal{G})$ and $\mathcal{E}(\mathcal{G}_2) \subseteq \mathcal{E}(\mathcal{G})$, $(\delta_{\mathcal{E}\mathcal{V}}(\mathcal{E}(\mathcal{G}_2)), \mathcal{E}(\mathcal{G}_2))$ is the smallest subgraph of \mathcal{G} whose edge set is $\mathcal{E}(\mathcal{G}_2)$, $(\mathcal{V}(\mathcal{G}_1), \epsilon_{\mathcal{V}\mathcal{E}}(\mathcal{V}(\mathcal{G}_1)))$ is the largest subgraph of \mathcal{G} whose vertex set is $\mathcal{V}(\mathcal{G}_1)$, $(\overline{\epsilon_{\mathcal{E}\mathcal{V}}(\mathcal{E}(\mathcal{G}_2))}, \overline{\mathcal{E}(\mathcal{G}_2)})$ is the smallest subgraph of \mathcal{G} whose edge set is $\overline{\mathcal{E}(\mathcal{G}_2)}$, and $(\overline{\mathcal{V}(\mathcal{G}_1)}, \overline{\delta_{\mathcal{V}\mathcal{E}}(\mathcal{V}(\mathcal{G}_1))})$ is the largest subgraph of \mathcal{G} whose vertex set is $\overline{\mathcal{V}(\mathcal{G}_1)}$.

These operators are illustrated in Figs. 1.1a-f. The following property locally characterizes them.

Property 4 ([10])

For any $\mathcal{E}(\mathcal{G}_1) \subseteq \mathcal{E}(\mathcal{G})$ and $\mathcal{V}(\mathcal{G}_2) \subseteq \mathcal{V}(\mathcal{G})$:

1. $\delta_{\mathcal{E}\mathcal{V}} : \mathcal{E}(\mathcal{G}) \rightarrow \mathcal{V}(\mathcal{G})$ is such that $\delta_{\mathcal{E}\mathcal{V}}(\mathcal{E}(\mathcal{G}_1)) = \{v_i \in \mathcal{V}(\mathcal{G}) \mid \exists e_{ij} \in \mathcal{E}(\mathcal{G}_1)\}$;
2. $\epsilon_{\mathcal{V}\mathcal{E}} : \mathcal{V}(\mathcal{G}) \rightarrow \mathcal{E}(\mathcal{G})$ is such that $\epsilon_{\mathcal{V}\mathcal{E}}(\mathcal{V}(\mathcal{G}_2)) = \{e_{ij} \in \mathcal{E}(\mathcal{G}) \mid v_i \in \mathcal{V}(\mathcal{G}_2) \text{ and } v_j \in \mathcal{V}(\mathcal{G}_2)\}$;
3. $\epsilon_{\mathcal{E}\mathcal{V}} : \mathcal{E}(\mathcal{G}) \rightarrow \mathcal{V}(\mathcal{G})$ is such that $\epsilon_{\mathcal{E}\mathcal{V}}(\mathcal{E}(\mathcal{G}_1)) = \{v_i \in \mathcal{V}(\mathcal{G}) \mid \forall e_{ij} \in \mathcal{E}(\mathcal{G}), e_{ij} \in \mathcal{E}(\mathcal{G}_1)\}$;

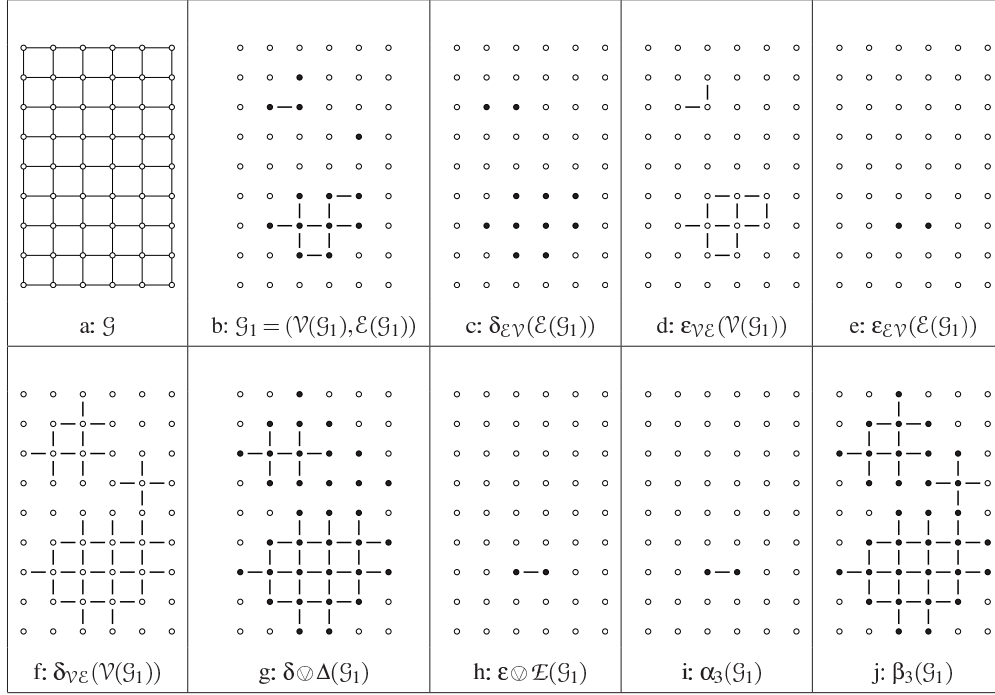


Figure 1.1: Dilations and erosions.

4. $\delta_{\mathcal{V}\mathcal{E}} : \mathcal{V}(\mathcal{G}) \rightarrow \mathcal{E}(\mathcal{G})$ is such that $\delta_{\mathcal{V}\mathcal{E}}(\mathcal{V}(\mathcal{G}_2)) = \{e_{ij} \in \mathcal{E}(\mathcal{G}) \mid \text{either } v_i \in \mathcal{V}(\mathcal{G}_2) \text{ or } v_j \in \mathcal{V}(\mathcal{G}_2)\}$.

In other words, $\delta_{\mathcal{E}\mathcal{V}}(\mathcal{E}(\mathcal{G}_1))$ is the set of all vertices which belong to an edge of $\mathcal{E}(\mathcal{G}_1)$, $\varepsilon_{\mathcal{V}\mathcal{E}}(\mathcal{V}(\mathcal{G}_2))$ is the set of all edges whose two extremities are in $\mathcal{V}(\mathcal{G}_2)$, $\varepsilon_{\mathcal{E}\mathcal{V}}(\mathcal{E}(\mathcal{G}_1))$ is the set of all vertices which do not belong to any edge of $\overline{\mathcal{E}(\mathcal{G}_1)}$, and $\delta_{\mathcal{V}\mathcal{E}}(\mathcal{V}(\mathcal{G}_2))$ is the set of all edges which have at least one extremity in $\mathcal{V}(\mathcal{G}_2)$.

From this characterization, we can recognize the general graph version of some operators introduced by Meyer and Angulo [8] (see also [7]) for the hexagonal grid. By translating supremum and infimum of the graph lattice to the corresponding supremum and infimum of the lattice of weights, we obtain the weighted version of these operators. The four basic operators of Definition 3 translate to: $(\delta_{\mathcal{E}\mathcal{V}}(w))_i = \max\{w_{ij} \mid e_{ij} \in \mathcal{E}\}$, $\varepsilon_{\mathcal{E}\mathcal{V}}(w)_i = \min\{w_{ij} \mid e_{ij} \in \mathcal{E}\}$, $(\delta_{\mathcal{V}\mathcal{E}}(w))_{ij} = \max\{w_i, w_j\}$ and $(\varepsilon_{\mathcal{V}\mathcal{E}}(w))_{ij} = \min\{w_i, w_j\}$.

Before further analyzing the operators defined above, let us briefly recall some algebraic tools which are fundamental in mathematical morphology [26].

Given two lattices \mathcal{L}_1 and \mathcal{L}_2 , an operator $\delta : \mathcal{L}_1 \rightarrow \mathcal{L}_2$ is called a *dilation* when it preserves the supremum (*i.e.* $\forall X \subseteq \mathcal{L}_1, \delta(\vee_1 X) = \vee_2 \{\delta(x) \mid x \in X\}$, where \vee_1 is the supremum in \mathcal{L}_1 and \vee_2 the supremum in \mathcal{L}_2). Similarly, an operator which preserves the infimum is called an *erosion*.

Two operators $\varepsilon : \mathcal{L}_1 \rightarrow \mathcal{L}_2$ and $\delta : \mathcal{L}_2 \rightarrow \mathcal{L}_1$ form an *adjunction* (ε, δ) when for any x in \mathcal{L}_2 and any y in \mathcal{L}_1 , we have $\delta(x) \leq_1 y \Leftrightarrow x \leq_2 \varepsilon(y)$, where \leq_1 and \leq_2 denote the order relations on respectively \mathcal{L}_1 and \mathcal{L}_2 . Given two operators ε and δ , if the pair (ε, δ) is an adjunction, then ε is an erosion and δ is a dilation.

Given two complemented lattices \mathcal{L}_1 and \mathcal{L}_2 , two operators α and β from \mathcal{L}_1 into \mathcal{L}_2 are *dual* (with respect to the complement) of each other when, for any $x \in \mathcal{L}_1$, we have $\beta(x) = \overline{\alpha(\bar{x})}$. If α and β are dual of each other, then β is an erosion whenever α is a dilation.

Property 5 (dilation, erosion, adjunction, duality, [10])

1. Both $(\epsilon_{\mathcal{V}\mathcal{E}}, \delta_{\mathcal{E}\mathcal{V}})$ and $(\epsilon_{\mathcal{E}\mathcal{V}}, \delta_{\mathcal{V}\mathcal{E}})$ are adjunctions.
2. Operators $\epsilon_{\mathcal{V}\mathcal{E}}$ and $\delta_{\mathcal{V}\mathcal{E}}$ (resp. $\epsilon_{\mathcal{E}\mathcal{V}}$ and $\delta_{\mathcal{E}\mathcal{V}}$) are dual of each other.
3. Operators $\delta_{\mathcal{E}\mathcal{V}}$ and $\delta_{\mathcal{V}\mathcal{E}}$ are dilations.
4. Operators $\epsilon_{\mathcal{E}\mathcal{V}}$ and $\epsilon_{\mathcal{V}\mathcal{E}}$ are erosions.

Let us compose these dilations and erosions to act on $\mathcal{V}(\mathbb{G})$ and $\mathcal{E}(\mathbb{G})$.

Definition 6 (vertex-dilation, vertex-erosion)

We define δ and ϵ that act on $\mathcal{V}(\mathbb{G})$ (i.e., $\mathcal{V}(\mathbb{G}) \rightarrow \mathcal{V}(\mathbb{G})$) by $\delta = \delta_{\mathcal{E}\mathcal{V}} \circ \delta_{\mathcal{V}\mathcal{E}}$ and $\epsilon = \epsilon_{\mathcal{E}\mathcal{V}} \circ \epsilon_{\mathcal{V}\mathcal{E}}$.

As compositions of respectively dilations and erosions, δ and ϵ are respectively a dilation and an erosion. Moreover, by composition of adjunctions and dual operators, δ and ϵ are dual and (ϵ, δ) is an adjunction.

In fact, it can be shown that δ and ϵ correspond exactly to the usual notions of an erosion and of a dilation of a set of vertices in a graph [4, 5]. It means, in particular that, when $\mathcal{V}(\mathcal{G})$ is a subset of the grid points \mathbb{Z}^d and when the edge set $\mathcal{E}(\mathcal{G})$ is obtained from a symmetrical structuring element, then the operators defined above are equivalent to the usual binary dilation and erosion by the considered structuring element. For instance, in Fig. 1.1, $\mathcal{V}(\mathcal{G})$ is a rectangular subset of \mathbb{Z}^2 and $\mathcal{E}(\mathcal{G})$ corresponds to the basic “cross” structuring element. It can be verified that the vertex sets in Fig. 1.1g and h, obtained by applying δ and ϵ to $\mathcal{V}(\mathcal{G}_1)$ (Fig. 1.1b), are the dilation and the erosion by a “cross” structuring element of $\mathcal{V}(\mathcal{G}_1)$.

We now consider a dual/adjunct pair of dilation and erosion acting on $\mathcal{E}(\mathbb{G})$.

Definition 7 (edge-dilation, edge-erosion, [10])

We define Δ and \mathcal{E} that act on $\mathcal{E}(\mathbb{G})$ by $\Delta = \delta_{\mathcal{V}\mathcal{E}} \circ \delta_{\mathcal{E}\mathcal{V}}$ and $\mathcal{E} = \epsilon_{\mathcal{V}\mathcal{E}} \circ \epsilon_{\mathcal{E}\mathcal{V}}$.

Definition 8 ([10])

We define the operators $\delta \odot \Delta$ and $\epsilon \odot \mathcal{E}$ by respectively $(\delta(\mathcal{V}(\mathcal{G}_1)), \Delta(\mathcal{E}(\mathcal{G}_1)))$ and $(\epsilon(\mathcal{V}(\mathcal{G}_1)), \mathcal{E}(\mathcal{E}(\mathcal{G}_1)))$, for any $\mathcal{G}_1 \in \mathbb{G}$.

For instance, Figs. 1.1f and 1.1g present the results obtained by applying the operator $\delta \odot \Delta$ and the operator $\epsilon \odot \mathcal{E}$ to the subgraph \mathcal{G}_1 (Fig. 1.1b) of \mathcal{G} (Fig. 1.1a).

Theorem 9 (graph-dilation, graph-erosion, [10])

The operators $\delta \odot \Delta$ and $\epsilon \odot \mathcal{E}$ are respectively a dilation and an erosion acting on the lattice $(\mathbb{G}, \sqsubseteq)$. Furthermore, $(\epsilon \odot \mathcal{E}, \delta \odot \Delta)$ is an adjunction.

Note that since lattice \mathbb{G} is sup-generated by set \mathcal{S} , it suffices to know the dilation of the graphs in \mathcal{S} for characterizing the dilation of the graphs in \mathbb{G} .

Compared to classical morphological operators on sets, the dilations and erosions introduced in this section furthermore convey some connectivity properties different than the ones which can be deduced from classical dilations and erosions. Observe, for instance, in Fig. 1.1g, that some 4-adjacent vertices of $\delta(\mathcal{V}(\mathcal{G}_1))$ are not linked by an edge in the graph $\delta \odot \Delta(\mathcal{G}_1)$. These properties can be useful in further processing involving for instance connected operators [27, 28, 29, 30].

Thanks to the operators presented in Definition 3, other intersecting adjunctions (hence dilations/erosions) can be defined on \mathbb{G} :

1. (α_1, β_1) such that $\forall \mathcal{G}_1 \in \mathbb{G}$, $\alpha_1(\mathcal{G}_1) = (\mathcal{V}(\mathcal{G}_1), \mathcal{E}(\mathcal{G}_1))$ and $\beta_1(\mathcal{G}_1) = (\delta_{\mathcal{E}\mathcal{V}}(\mathcal{E}(\mathcal{G}_1)), \mathcal{E}(\mathcal{G}_1))$;
2. (α_2, β_2) such that $\forall \mathcal{G}_1 \in \mathbb{G}$, $\alpha_2(\mathcal{G}_1) = (\mathcal{V}(\mathcal{G}_1), \mathcal{E}_{\mathcal{V}\mathcal{E}}(\mathcal{V}(\mathcal{G}_1)))$ and $\beta_2(\mathcal{G}_1) = (\mathcal{V}(\mathcal{G}_1), \emptyset)$;
3. (α_3, β_3) such that $\forall \mathcal{G}_1 \in \mathbb{G}$, $\alpha_3(\mathcal{G}_1) = (\mathcal{E}_{\mathcal{E}\mathcal{V}}(\mathcal{E}(\mathcal{G}_1)), \mathcal{E}_{\mathcal{V}\mathcal{E}} \circ \mathcal{E}_{\mathcal{E}\mathcal{V}}(\mathcal{E}(\mathcal{G}_1)))$
and $\beta_3(\mathcal{G}_1) = (\delta_{\mathcal{E}\mathcal{V}} \circ \delta_{\mathcal{V}\mathcal{E}}(\mathcal{V}(\mathcal{G}_1)), \delta_{\mathcal{V}\mathcal{E}}(\mathcal{V}(\mathcal{G}_1)))$.

The adjunction (α_3, β_3) is illustrated in Fig. 1.1i and 1.1j. Note also that, using usual graph terminologies, β_1 (resp. α_2) can be defined as the operator which associates to a graph the graph induced by its edge set (resp. vertex set).

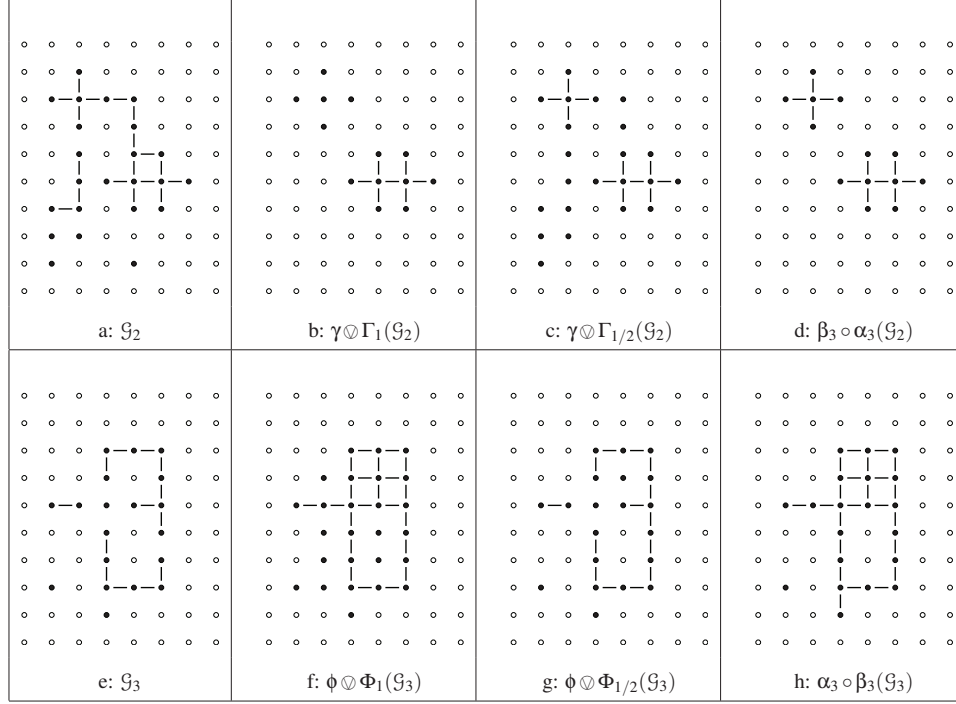
A more general version of those operators can be obtained in the framework of complexes [12], allowing us to deal for example with meshes. There also exist other recent graph-based approaches to morphology, for example based on hypergraphs [11] or on discrete differential equations [9].

1.4 Filters

In mathematical morphology, a *filter* is an operator α acting on a lattice \mathcal{L} , which is increasing (*i.e.* $\forall x, y \in \mathcal{L}$, $\alpha(x) \leq \alpha(y)$ whenever $x \leq y$) and idempotent (*i.e.* $\forall x \in \mathcal{L}$, $\alpha(\alpha(x)) = \alpha(x)$). A filter α on \mathcal{L} which is extensive (*i.e.* $\forall x \in \mathcal{L}$, $x \leq \alpha(x)$) is called a *closing on \mathcal{L}* whereas a filter α on \mathcal{L} which is anti-extensive (*i.e.* $\forall x \in \mathcal{L}$, $\alpha(x) \leq x$) is called an *opening on \mathcal{L}* . It is known that composing the two operators of an adjunction yields an opening or a closing depending on the order in which the operators are composed [26]. In this section, the operators of Section 1.3 are composed to obtain filters on $\mathcal{V}(\mathbb{G})$, $\mathcal{E}(\mathbb{G})$ and \mathbb{G} .

Definition 10 (opening, closing, [10])

1. We define γ_1 and ϕ_1 , that act on $\mathcal{V}(\mathbb{G})$, by $\gamma_1 = \delta \circ \varepsilon$ and $\phi_1 = \varepsilon \circ \delta$.
2. We define Γ_1 and Φ_1 , that act on $\mathcal{E}(\mathbb{G})$, by $\Gamma_1 = \Delta \circ \mathcal{E}$ and $\Phi_1 = \mathcal{E} \circ \Delta$.

Figure 1.2: Openings and closings (\mathcal{G} is induced by the 4-adjacency relation).

3. We define the operators $\gamma \odot \Gamma_1$ and $\phi \odot \Phi_1$ by respectively $\gamma \odot \Gamma_1(\mathcal{G}_1) = (\gamma_1(\mathcal{V}(\mathcal{G}_1)), \Gamma_1(\mathcal{E}(\mathcal{G}_1)))$ and $\phi \odot \Phi_1(\mathcal{G}_1) = (\phi_1(\mathcal{V}(\mathcal{G}_1)), \Phi_1(\mathcal{E}(\mathcal{G}_1)))$ for any $\mathcal{G}_1 \in \mathbb{G}$.

Figs. 1.2b and 1.2f present the result of $\gamma \odot \Gamma_1$ and $\phi \odot \Phi_1$ for respectively the subgraph of Fig. 1.2a and the one of Fig. 1.2e.

The opening γ_1 and the closing ϕ_1 correspond to the classical opening and closing on the vertices. The closing Γ_1 and the opening Φ_1 are the corresponding edge-version of these operators. By combination, we obtain $\gamma \odot \Gamma_1$ and $\phi \odot \Phi_1$.

In fact, by composing $\delta_{\mathcal{E}\mathcal{V}}$ with $\epsilon_{\mathcal{V}\mathcal{E}}$ and $\delta_{\mathcal{E}\mathcal{V}}$ with $\epsilon_{\mathcal{V}\mathcal{E}}$, we obtain smaller filters.

Definition 11 (half-opening, half-closing, [10])

1. We define $\gamma_{1/2}$ and $\phi_{1/2}$, that act on $\mathcal{V}(\mathbb{G})$, by $\gamma_{1/2} = \delta_{\mathcal{E}\mathcal{V}} \circ \epsilon_{\mathcal{V}\mathcal{E}}$ and $\phi_{1/2} = \epsilon_{\mathcal{E}\mathcal{V}} \circ \delta_{\mathcal{V}\mathcal{E}}$.
2. We define $\Gamma_{1/2}$ and $\Phi_{1/2}$, that act on $\mathcal{E}(\mathbb{G})$ by $\Gamma_{1/2} = \delta_{\mathcal{V}\mathcal{E}} \circ \epsilon_{\mathcal{E}\mathcal{V}}$ and $\Phi_{1/2} = \epsilon_{\mathcal{V}\mathcal{E}} \circ \delta_{\mathcal{E}\mathcal{V}}$.
3. We define the operators $\gamma \odot \Gamma_{1/2}$ and $\phi \odot \Phi_{1/2}$ by respectively $\gamma \odot \Gamma_{1/2}(\mathcal{G}_1) = (\gamma_{1/2}(\mathcal{V}(\mathcal{G}_1)), \Gamma_{1/2}(\mathcal{E}(\mathcal{G}_1)))$ and $\phi \odot \Phi_{1/2}(\mathcal{G}_1) = (\phi_{1/2}(\mathcal{V}(\mathcal{G}_1)), \Phi_{1/2}(\mathcal{E}(\mathcal{G}_1)))$, for any $\mathcal{G}_1 \in \mathbb{G}$.

Thanks to Property 4, the operators defined above can be locally characterized. Let $\mathcal{V}(\mathcal{G}_1) \subseteq \mathcal{V}(\mathcal{G})$ and $\mathcal{E}(\mathcal{G}_2) \subseteq \mathcal{E}(\mathcal{G})$, we have:

$$\begin{aligned}
\gamma_{1/2}(\mathcal{V}(\mathcal{G}_1)) &= \{v_i \in \mathcal{V}(\mathcal{G}_1) \mid \exists e_{ij} \in \mathcal{E}(\mathcal{G}) \text{ with } v_j \in \mathcal{V}(\mathcal{G}_1)\} \\
&= \mathcal{V}(\mathcal{G}_1) \setminus \{v_i \in \mathcal{V}(\mathcal{G}_1) \mid \forall e_{ij} \in \mathcal{E}(\mathcal{G}), v_j \notin \mathcal{V}(\mathcal{G}_1)\} \\
\Gamma_{1/2}(\mathcal{E}(\mathcal{G}_2)) &= \{e_{ij} \in \mathcal{E}(\mathcal{G}) \mid \{e_{ik} \in \mathcal{E}(\mathcal{G})\} \subseteq \mathcal{E}(\mathcal{G}_2)\} \\
&= \mathcal{E}(\mathcal{G}_2) \setminus \{e \in \mathcal{E}(\mathcal{G}) \mid \forall v_i \in e, \exists e_{ij} \in \mathcal{E}(\mathcal{G}) \text{ with } e_{ij} \notin \mathcal{E}(\mathcal{G}_2)\} \\
\phi_{1/2}(\mathcal{V}(\mathcal{G}_1)) &= \{v_i \in \mathcal{V}(\mathcal{G}) \mid \text{either } v_i \in \mathcal{V}(\mathcal{G}_1) \text{ or } \forall e_{ij} \in \mathcal{E}(\mathcal{G}), v_j \in \mathcal{V}(\mathcal{G}_1)\} \\
&= \mathcal{V}(\mathcal{G}_1) \cup \{v_i \in \overline{\mathcal{V}(\mathcal{G}_1)} \mid \forall e_{ij} \in \mathcal{E}(\mathcal{G}), v_j \in \mathcal{V}(\mathcal{G}_1)\} \\
\Phi_{1/2}(\mathcal{E}(\mathcal{G}_2)) &= \{e_{ij} \in \mathcal{E}(\mathcal{G}) \mid \exists e_{ik} \in \mathcal{E}(\mathcal{G}_2) \text{ and } \exists e_{jl} \in \mathcal{E}(\mathcal{G}_2)\} \\
&= \mathcal{E}(\mathcal{G}_2) \cup \{e_{ij} \in \overline{\mathcal{E}(\mathcal{G}_2)} \mid v_i \in \delta_{\mathcal{E}(\mathcal{G}_2)} \text{ and } v_j \in \delta_{\mathcal{E}(\mathcal{G}_2)}\}.
\end{aligned}$$

Informally speaking, $\gamma_{1/2}$ removes from \mathcal{G}_1 its isolated vertices whereas $\Gamma_{1/2}$ removes from \mathcal{G}_2 the edges which do not contain a vertex completely covered by edges in \mathcal{G}_2 . It may be furthermore seen that $\gamma_{1/2}$ (resp. $\Gamma_{1/2}$) is the dual of $\phi_{1/2}$ (resp. $\Phi_{1/2}$). Thus, $\phi_{1/2}$ adds to \mathcal{G}_1 the vertices of $\overline{\mathcal{V}(\mathcal{G}_1)}$ completely surrounded by elements of \mathcal{G}_1 whereas $\Phi_{1/2}$ adds to \mathcal{G}_2 the edges of $\overline{\mathcal{E}(\mathcal{G}_2)}$ whose two extremities belong to at least one edge in \mathcal{G}_2 (see for instance Fig. 1.2).

The family \mathbb{G} is closed under the operators presented in Definition 10.3 since they are obtained by composition of operators also satisfying this property. Furthermore, it can be deduced from the local characterization of the operators $\gamma_{1/2}$, $\Gamma_{1/2}$, $\phi_{1/2}$ and $\Phi_{1/2}$ that the family \mathbb{G} is also closed under the operators of Definition 11.3. Hence, thanks to the properties of adjunctions recalled in the introduction of this section, the following theorem can be established.

Theorem 12 (graph-openings, graph-closings,[10])

1. The operators $\gamma_{1/2}$ and γ_1 (resp. $\Gamma_{1/2}$ and Γ_1) are openings on $\mathcal{V}(\mathbb{G})$ (resp. $\mathcal{E}(\mathbb{G})$) and $\phi_{1/2}$, and Φ_1 (resp. $\Phi_{1/2}$ and ϕ_1) are closings on $\mathcal{V}(\mathbb{G})$ (resp. $\mathcal{E}(\mathbb{G})$).
2. The family \mathbb{G} is closed under $\gamma \odot \Gamma_{1/2}$, $\phi \odot \Phi_{1/2}$, $\gamma \odot \Gamma_1$, and $\phi \odot \Phi_1$.
3. The operators $\gamma \odot \Gamma_{1/2}$ and $\gamma \odot \Gamma_1$ are openings on \mathbb{G} and $\phi \odot \Phi_{1/2}$ and $\phi \odot \Phi_1$ are closings on \mathbb{G} .

Composing the operators of the adjunctions (α_i, β_i) , defined at the end of Section 1.3, also yields remarkable openings and closings. Indeed, it can be easily seen that: $\alpha_1 \circ \beta_1 = \alpha_1$, $\alpha_2 \circ \beta_2 = \alpha_2$, $\beta_1 \circ \alpha_1 = \beta_1$ and $\beta_2 \circ \alpha_2 = \beta_2$. Thus α_1 and α_2 are both a closing and an erosion and β_1 and β_2 are both a dilation and an opening. This means, in particular, that α_1 and α_2 are idempotent extensive erosions and that β_1 and β_2 are idempotent anti-extensive dilations. The opening and the closing resulting from the adjunction (α_3, β_3) are illustrated in Figs. 1.2d and 1.2h.



Figure 1.3: Illustration of classical versus vertex-edge alternating filter (see [10]).

The weighted versions of those various operators also have an interpretation related to classical notions of graph theory. Consider for example the weighted version of the opening $\Gamma_{1/2} = \delta_{V\mathcal{E}}\varepsilon_{\mathcal{E}V}$. The dilation $\delta_{V\mathcal{E}}$ assigns to each edge the highest weight of its adjacent vertices; but each adjacent vertex has been assigned by $\varepsilon_{\mathcal{E}V}$ the weight of its lowest edge. So if w_{ij} is left unchanged by $\Gamma_{1/2}$, it means that the edge e_{ij} is (one of) the lowest edges of v_i or of v_j , say v_i . Its weight is higher or equal to the weight of the lowest edge of v_j . Thus, using some well-known properties of minimum spanning trees (see Section 1.6.3 for a precise definition of a minimum spanning tree), it can be shown that the graph induced by the set of edges that are left invariant by $\Gamma_{1/2}$ is the union of all minimum spanning trees of this graph, closely related to the *Gabriel graph* [31].

It is possible to associate with any lattice \mathcal{L} , the lattice of all increasing operators on \mathcal{L} . In this context, two filters φ_1 and φ_2 on the lattice \mathcal{L} are said *ordered* if, for any $x \in \mathcal{L}$, $\varphi_1(x) \leq \varphi_2(x)$ or if, for any $x \in \mathcal{L}$, $\varphi_2(x) \leq \varphi_1(x)$. A usual way to build a hierarchy of filters (*i.e.* an ordered family of filters) from an adjunct pair (α, β) of erosion and dilation consists of building the dilations and erosions obtained by iterating several times α and β . In general, composing these iterated versions of α and β leads to hierarchies of filters when the number of iterations increases. By combining and iterating the operators that we have defined in this section, we obtain [10] hierarchies of filters in the lattice \mathbb{G} that perform better than the classical ones, in the sense that they are able to remove more noise (see Fig. 1.3.)

1.5 Connected operators and filtering with the component tree

Gray-level connected operators [27] act by merging neighboring “flat” zones. They cannot create new contours and, as a result, they cannot introduce in the output image a structure that is not present in the input image. Furthermore, they cannot modify the position of existing boundaries between regions, and therefore have very good contour preservation properties.

To create “flat” zones, a simple operation on a weighted graph is thresholding, that produces a level set. For $\lambda \in \mathbb{R}^+$, the level-sets of an edge-weighted graph are denoted $w_{\mathcal{E}}^{\leq}[\lambda] = \{e_{ij} \in \mathcal{E} | w_{ij} \leq \lambda\}$. We define $\mathcal{C}_{\mathcal{E}}^{\leq}$

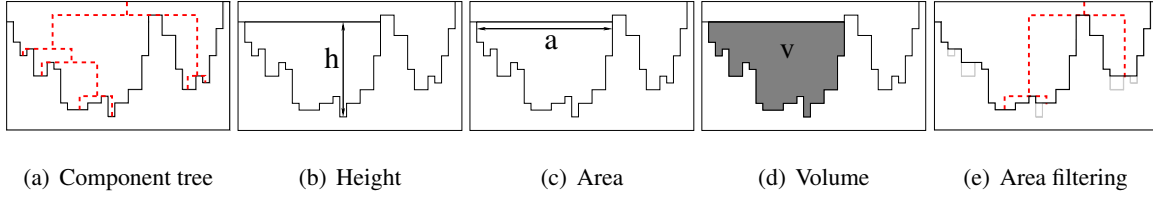


Figure 1.4: Illustration of respectively a component tree (dashed lines), the height, the area and the volume of a component and an area filtering.

as the set composed of all the pairs $[\lambda, C]$, where $\lambda \in \mathbb{R}^+$ and C is a (connected) component of the graph $w_\varepsilon^\leq[\lambda]$. We note that one can reconstruct w from $\mathcal{C}_\varepsilon^\leq$; more precisely, we have:

$$w_{ij} = \min\{\lambda \mid [\lambda, C] \in \mathcal{C}_\varepsilon^\leq, e_{ij} \in \mathcal{E}(C)\} \quad (1.1)$$

One can remark that two elements of $\mathcal{C}_\varepsilon^\leq$ are either disjoint or nested, thus it is easy to order the elements of $\mathcal{C}_\varepsilon^\leq$ in a tree. This tree is called the (*min*) *component tree*, and there exists fast algorithms to compute it [32, 33].

Using $\mathcal{C}_\varepsilon^\leq$ allows us to deal with specific components of w . For example, a minimum of w is a component C such that there exists λ with $[\lambda, C] \in \mathcal{C}_\varepsilon^\leq$ and it does not exist $[\lambda_1, C_1] \in \mathcal{C}_\varepsilon^\leq$ with $C_1 \sqsubset C$ and $C_1 \neq C$. Observe that a minimum of w is a graph. The set of all minima of w is denoted \mathcal{M} . Similarly, we can define $\mathcal{C}_\varepsilon^\geq$ allowing us to deal with maxima of edge-weighted graphs, $\mathcal{C}_\varepsilon^\leq$ and $\mathcal{C}_\varepsilon^\geq$ allowing us to deal with node-weighted graphs, and $\mathcal{C}_{\varepsilon\mathcal{E}}^\leq$ and $\mathcal{C}_{\varepsilon\mathcal{E}}^\geq$ allowing us to deal with node-and-edge weighted graphs.

We mention that other trees are possible, for example the binary partition tree [34], the tree of level lines also known as the inclusion tree or the tree of shapes [35], etc. The interest of the tree of shapes is that it allows us to interact both with maxima and minima, in a self-dual manner. See [36] for a survey of the usage of those trees in image processing. Here, we illustrate the usage of the component tree for filtering.

Given Eq. (1.1), filtering a graph can be seen as equivalent to removing some components of (say) $\mathcal{C}_\varepsilon^\leq$. Such a filtering is called *flooding*. One way to make this idea practicable is to design an attribute that tells us if a given component should be kept or not. Among the numerous attributes that can be computed, three are natural: the height, the area, and the volume (Fig. 1.4). Let $[\lambda, C] \in \mathcal{C}_\varepsilon^\leq$. We define

$$\text{height}([\lambda, C]) = \max\{\lambda - w_{ij} \mid e_{ij} \in \mathcal{E}(C)\} \quad (1.2)$$

$$\text{area}([\lambda, C]) = |\mathcal{V}(C)| \quad (1.3)$$

$$\text{volume}([\lambda, C]) = \sum_{e_{ij} \in \mathcal{E}(C)} (\lambda - w_{ij}) \quad (1.4)$$

For example, removing all the components whose area is lower than a threshold is a closing of w (Fig. 1.4.e). Remark that a filtering with either the height or the volume is not a closing, as such a filtering is not idempotent.

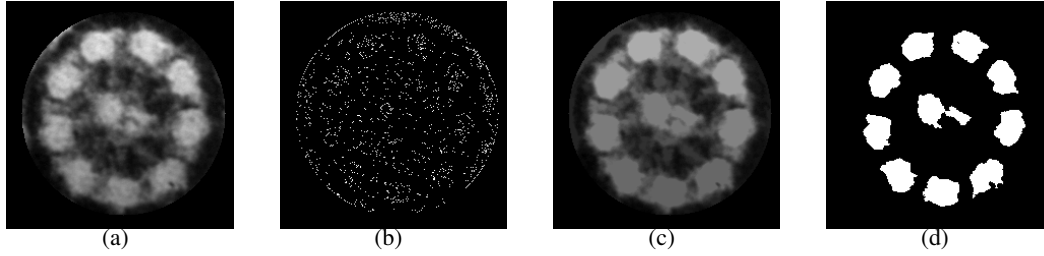


Figure 1.5: (a) Original image. (b) Maxima of image (a), in white. (c) Filtered image. (d) Maxima of image (c), which correspond to the ten most significant lobes of the image (a).

Fig. 1.5 illustrates this kind of filtering. Fig. 1.5.a is an image of a cell, in which we want to extract the ten bright lobes. If we consider that the brighter a pixel is, the higher its weight, then the tree we want to process is $\mathcal{C}_V^>$. Fig. 1.5.b shows that the image 1.5.a contains numerous maxima. Fig. 1.5.c is the filtered image obtained by a volume-based filtering, and Fig. 1.5.d shows the maxima of this filtered image.

As the components of the level sets can be ordered into a tree, they form a hierarchy of components. In the context of classification [37], hierarchies are widely studied, and the morphological framework can shed some light on the tools that are developed in that community. For example, let us look at the application Ψ that associates to any set of edge-weights w the map $\Psi(w)$ such that for any edge $e_{ij} \in \mathcal{E}$, $\Psi(w)(e_{ij}) = \min\{\lambda \mid [\lambda, C] \in \mathcal{C}_\mathcal{E}^<, v_i \in \mathcal{V}(C), v_j \in \mathcal{V}(C)\}$. Intuitively, and using a geographical metaphore, $\Psi(w)(e_{ij})$ is the lowest altitude to which one has to climb to go from v_i to v_j . It is straightforward to see that $\Psi(w) \leq w$, that $\Psi(\Psi(w)) = \Psi(w)$ and that if $w' \leq w$, $\Psi(w') \leq \Psi(w)$. Thus Ψ is an opening [38]. We remark that the subset of strictly positive maps that are defined on the complete graph $(\mathcal{V}, \mathcal{V} \times \mathcal{V})$ and that are open with respect to Ψ is the set of ultrametric distances on \mathcal{V} . The mapping Ψ is known under several names, including “subdominant ultrametric” and “ultrametric opening”. It is well known that Ψ is associated to the simplest method for hierarchical classification called single linkage clustering [39, 40], closely related to Kruskal’s algorithm [41] for computing a minimum spanning tree.

Before concluding the section, let us mention another flooding that has been used as a preprocessing step for watershed segmentation with markers. It is called *geodesic reconstruction from the markers* [42, 43], where the markers is a subgraph. It consists in removing all the components of $\mathcal{C}_\mathcal{E}^<$ that are not marked (*i.e.* that do not contain a vertex of the markers) and is given as a function w_R such that, for every edge e , we set $w_R(e)$ to be equal to the level λ of the lowest component of w containing e and at least one vertex of the markers. Observe that any component of w_R indeed contains at least one vertex of the markers. Given some markers, the geodesic reconstruction is a closing. Such a closing is also useful as an efficient preprocessing for a fast power-watershed optimization algorithm (see [15] and Section 1.8).

1.6 Watershed cuts

There exist many possible ways for defining a watershed [24, 44, 45, 46, 13, 47]. Intuitively, the watershed of a function (seen as a topographical surface) is composed of the locations from which a drop of water could flow towards different minima. The framework of edge-weighted graph allows the formalization of this principle and the proof of several remarkable properties [13] that we review in the sequel of the section. We first show that watershed cuts can be equivalently defined by their “catchment basins” (through a steepest descent property) or by the “dividing lines” separating these catchment basins (through the drop of water principle). As far as we know, in discrete frameworks, a similar property does not hold. The second property establishes the optimality of watershed-cuts: there is an equivalence between the watershed-cuts and the separations induced by a minimum spanning forest relative to the minima.

1.6.1 The drop of water principle

Definition 13 (from Def. 12 in [46])

Let \mathcal{G}_1 and \mathcal{G}_2 be two non-empty subgraphs of \mathcal{G} . We say that \mathcal{G}_2 is an extension of \mathcal{G}_1 (in \mathcal{G}) if $\mathcal{G}_1 \subseteq \mathcal{G}_2$ and if any component of \mathcal{G}_2 contains exactly one component of \mathcal{G}_1 .

The notion of extension is very general. Many segmentation algorithms iteratively extend some seed components in a graph: they produce an extension of the seeds. Most of them terminate once they have reached an extension which cover all the vertices of the graph. The separation which is thus produced is called a graph cut. Let $\mathcal{S} \subseteq \mathcal{E}$. We denote by $\bar{\mathcal{S}}$ the complementary set of \mathcal{S} in \mathcal{E} , i.e., $\bar{\mathcal{S}} = \mathcal{E} \setminus \mathcal{S}$. Recall that the graph induced by \mathcal{S} , given by the dilation β_1 , is the graph whose edge set is \mathcal{S} and whose vertex set is made of all points which belong to an edge in \mathcal{S} , i.e., $(\{v \in \mathcal{V} \mid \exists e \in \mathcal{S}, v \in e\}, \mathcal{S})$. In the following, the graph induced by \mathcal{S} is also denoted by \mathcal{S} .

Definition 14

Let $\mathcal{G}_1 \subseteq \mathcal{G}$ and $\mathcal{S} \subseteq \mathcal{E}$. We say that \mathcal{S} is a (graph) cut for \mathcal{G}_1 if $\bar{\mathcal{S}}$ is an extension of \mathcal{G}_1 and if \mathcal{S} is minimal for this property, i.e., if $\mathcal{T} \subseteq \mathcal{S}$ and $\bar{\mathcal{T}}$ is an extension of \mathcal{G}_1 , then we have $\mathcal{T} = \mathcal{S}$.

We introduce the watershed cuts of an edge-weighted graph. To this end, we formalize the drop of water principle. Intuitively, the catchment basins constitute an extension of the minima and they are separated by “lines” from which a drop of water can flow down towards distinct minima.

Let $\pi = (v_0, \dots, v_n)$ be a path. The path π is *descending* if, for any $i \in [1, n-1]$, $w(v_{i-1}, v_i) \geq w(v_i, v_{i+1})$.

Definition 15 (drop of water principle, [13])

Let $\mathcal{S} \subseteq \mathcal{E}$. We say that \mathcal{S} is a watershed cut (or simply a watershed) if $\bar{\mathcal{S}}$ is an extension of \mathcal{M} and if for any $e_{ij} = \{v_i, v_j\} \in \mathcal{S}$, there exist $\pi_1 = (v_0^1 = v_i, \dots, v_n^1)$ and $\pi_2 = (v_0^2 = v_j, \dots, v_m^2)$ which are two descending

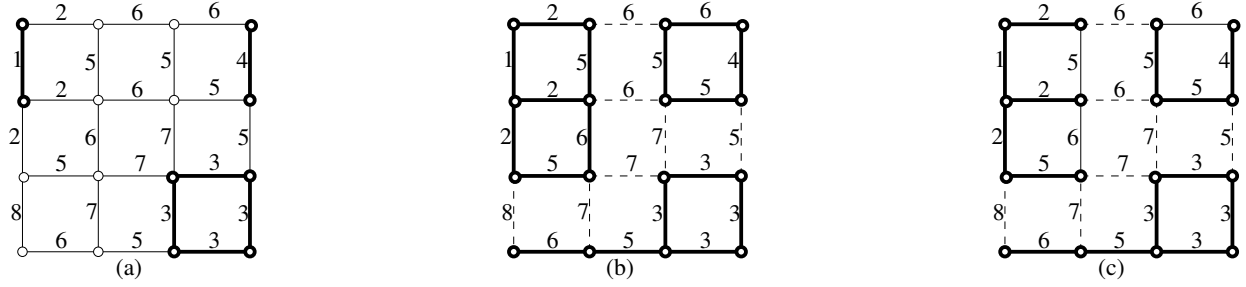


Figure 1.6: An edge-weighted graph \mathcal{G} . The edges and vertices drawn in bold are: (a), the minima of w ; (b), an extension of \mathcal{M} ; (c), an MSF rooted in \mathcal{M} . In (b), the set of dashed edges is a watershed cut of w .

paths in $\bar{\mathcal{S}}$ such that v_n^1 and v_m^2 are vertices of two distinct minima of w ; and $w_{ij} \geq w(\{v_0^1, v_1^1\})$ (resp. $w_{ij} \geq w(\{v_0^2, v_1^2\})$), whenever π_1 (resp. π_2) is not trivial.

We illustrate the previous definition on the edge-weighted graph depicted in Fig. 1.6. The weights w contain three minima (in bold in Fig. 1.6a). Let us denote by $\bar{\mathcal{S}}$ the set of dashed edges depicted in Fig. 1.6b and by $e = \{v_1, v_2\}$ the only edge whose altitude is 8. It may be seen that $\bar{\mathcal{S}}$ (in bold Fig. 1.6b) is an extension of \mathcal{M} . We also remark that there exists π_1 (resp. π_2) a descending path in $\bar{\mathcal{S}}$ from v_1 (resp. v_2) to the minimum at altitude 1 (resp. 3). The altitude of the first edge of π_1 (resp. π_2) is lower than the altitude of e . It can be verified that the previous properties hold true for any edge in $\bar{\mathcal{S}}$. Thus, $\bar{\mathcal{S}}$ is a watershed of F .

1.6.2 Catchment basins by a steepest descent property

A popular alternative to Definition 15 consists of defining a watershed exclusively by its catchment basins and the paths of steepest descent of w [48, 44, 49, 45] and does not involve any property of the divide. The following theorem (17) establishes the consistency of watershed cuts in edge-weighted graphs: they can be equivalently defined by a steepest descent property on the catchment basins (regions) or by the drop of water principle on the cut (border) which separate them. As far as we know, there is no definition of watershed in vertex-weighted graphs that verifies a similar property. This theorem thus emphasizes that the framework of edge-weighted graphs is adapted for the study of discrete watersheds.

Let $\pi = (v_0, \dots, v_l)$ be a path in \mathcal{G} . The path π is a *path of steepest descent* (for w) if, for any $i \in [1, l]$, $w(v_{i-1}, v_i) = (\varepsilon_{\mathcal{E}} v w)_{i-1} = \min\{w_{ij} | \forall j \text{ such that } e_{ij} \in \mathcal{E}\}$.

Definition 16 ([13])

Let $\mathcal{S} \subseteq \mathcal{E}$ be a cut for \mathcal{M} . We say that \mathcal{S} is a *basin cut* (of w) if, from each point of \mathcal{V} to \mathcal{M} , there exists, in the graph induced by $\bar{\mathcal{S}}$, a path of steepest descent for w . If \mathcal{S} is a basin cut of w , any component of $\bar{\mathcal{S}}$ is called a *catchment basin* (of w , for \mathcal{S}).

Theorem 17 (consistency, [13])

Let $S \subseteq \mathcal{E}$ be a cut for \mathcal{M} . The set S is a watershed if and only if S is a basin cut of w .

1.6.3 Minimum spanning forests

We establish the optimality of watersheds. To this end, we introduce the notion of minimum spanning forests rooted in some subgraphs of \mathcal{G} . Each of these forests induces a unique graph cut. The main result of this study (Th. 19) states that a graph cut is induced by a minimum spanning rooted in the minima of a function if and only if it is a watershed of this function.

Definition 18 (rooted MSF, [14])

Let \mathcal{G}_1 and \mathcal{G}_2 be two nonempty subgraphs of \mathcal{G} . We say that \mathcal{G}_2 is rooted in \mathcal{G}_1 if $\mathcal{V}(\mathcal{G}_1) \subseteq \mathcal{V}(\mathcal{G}_2)$ and if the vertex set of any component of \mathcal{G}_2 contains the vertex set of exactly one component of \mathcal{G}_1 . Recall that the weight of \mathcal{G}_1 is the sum of its weight, i.e. $\sum_{e_{ij} \in \mathcal{E}(\mathcal{G}_1)} w_{ij}$

We say that \mathcal{G}_2 is a minimum spanning forest (MSF) rooted in \mathcal{G}_1 if (1) \mathcal{G}_2 is spanning for \mathcal{V} , (2) \mathcal{G}_2 is rooted in \mathcal{G}_1 , and (3) the weight of \mathcal{G}_2 is less than or equal to the weight of any graph \mathcal{G}_3 satisfying (1) and (2) (i.e., \mathcal{G}_3 is both spanning for \mathcal{V} and rooted in \mathcal{G}_1). The set of all minimum spanning forests rooted in \mathcal{G}_1 is denoted by $MSF(\mathcal{G}_1)$.

The above definition of rooted MSFs allows the usual notions of graph theory based on trees and forests to be recovered. In particular, if v is a vertex of V , it can be seen that any element in $MSF(\{v\}, \emptyset)$ is a *minimum spanning tree* of (G, w) , and that, conversely, any minimum spanning tree of (G, w) belongs to $MSF(\{v\}, \emptyset)$.

Let us consider the edge-weighted graph \mathcal{G} and the subgraph \mathcal{M} (in bold) of Fig. 1.6a. It can be verified that the graph \mathcal{G}_1 (in bold in Fig. 1.6c) is an MSF rooted in \mathcal{M} .

We now have the mathematical tools to present the main result of this section (Th. 19) which establishes the optimality of watersheds.

Let \mathcal{G}_1 be a subgraph of \mathcal{G} and let \mathcal{G}_2 be a spanning forest rooted in \mathcal{G}_1 . There exists a unique cut for \mathcal{G}_2 and this cut is also a cut for \mathcal{G}_1 . We say that this unique cut is the *cut induced by \mathcal{G}_2* .

Theorem 19 (optimality, [13])

Let $S \subseteq \mathcal{E}$. The set S is a cut induced by a MSF rooted in \mathcal{M} if and only if S is a watershed cut.

1.6.4 Illustrations to segmentation

In this subsection, we illustrate the versatility of the proposed framework to perform segmentation on different kinds of geometric objects. Firstly, we show how to segment triangulated surfaces by watershed cuts, and

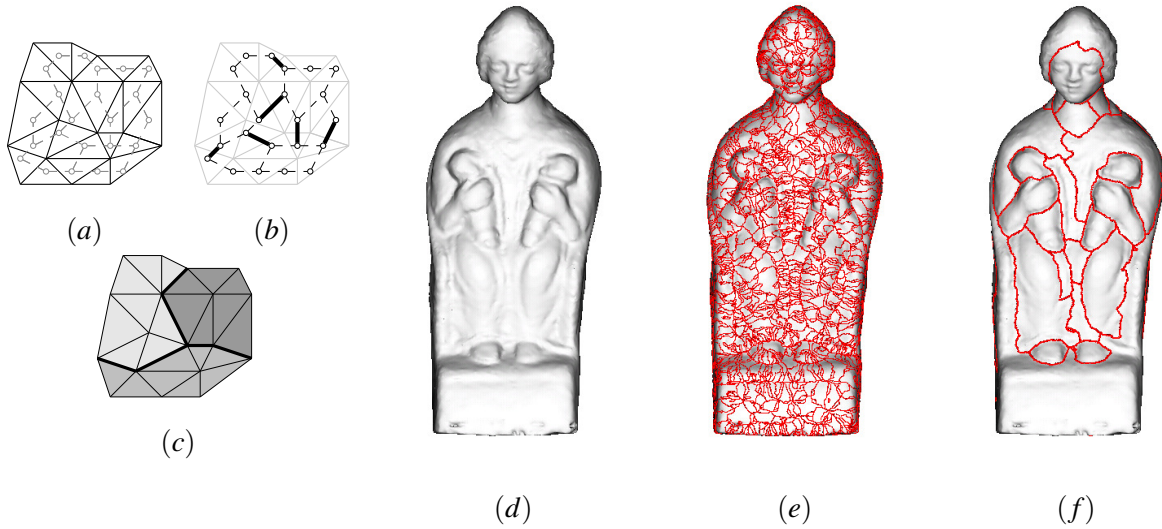


Figure 1.7: Surface segmentation by watershed cut. (a): A mesh in black and its associated graph in gray. (b): A cut on this graph (in bold); and (c), the corresponding segmentation of the mesh. (d): Rendering of the mesh of a sculpture. (e): A watershed (in red) of a map F which behaves like the inverse of the mean curvature and, in (f), a watershed of a filtered version of F . The mesh shown in (d) is provided by the French Museum Center for Research.

secondly we apply the watershed cuts to the segmentation of diffusion tensors images, which are medical images associating a tensor to each voxel.

3D shape acquisition and digitizing have received more and more attention for a decade, leading to an increasing amount of 3D surface-models (or meshes) such as the one in Fig. 1.7d. In a recent work [50], a new search engine has been proposed for indexing and retrieving objects of interests in a database of meshes (EROS 3D) provided by the French Museum Center for Research. One key idea of this search engine is to use region descriptors rather than global shape descriptors. In order to produce such descriptors, it is then essential to obtain meaningful mesh segmentations.

Informally, a *mesh* M in the 3D Euclidean space is a set of *triangles*, *sides* of triangles and *points* such that each side is included in exactly two triangles (see Fig. 1.7a). In order to perform a watershed cut on such a mesh, we build a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ whose vertex set \mathcal{V} is the set of all triangles in M and whose edge set \mathcal{E} is composed by the pairs $e_{ij} = \{v_i, v_j\}$ such that v_i and v_j are two triangles of M that share a common side (see Fig. 1.7a). The graph \mathcal{G} is known under the name of *2-dual of the surface mesh* [51].

To obtain a segmentation of the mesh M thanks to a watershed cut, we need to weight the edges of \mathcal{G} (or equivalently the sides of M) by a map whose values are high around the boundaries of the regions that we want to separate. We have found that the interesting contours on the EROS 3D meshes are mostly located on concave zones. Therefore, we weight the edges of \mathcal{G} by a weighting w which behaves like the inverse

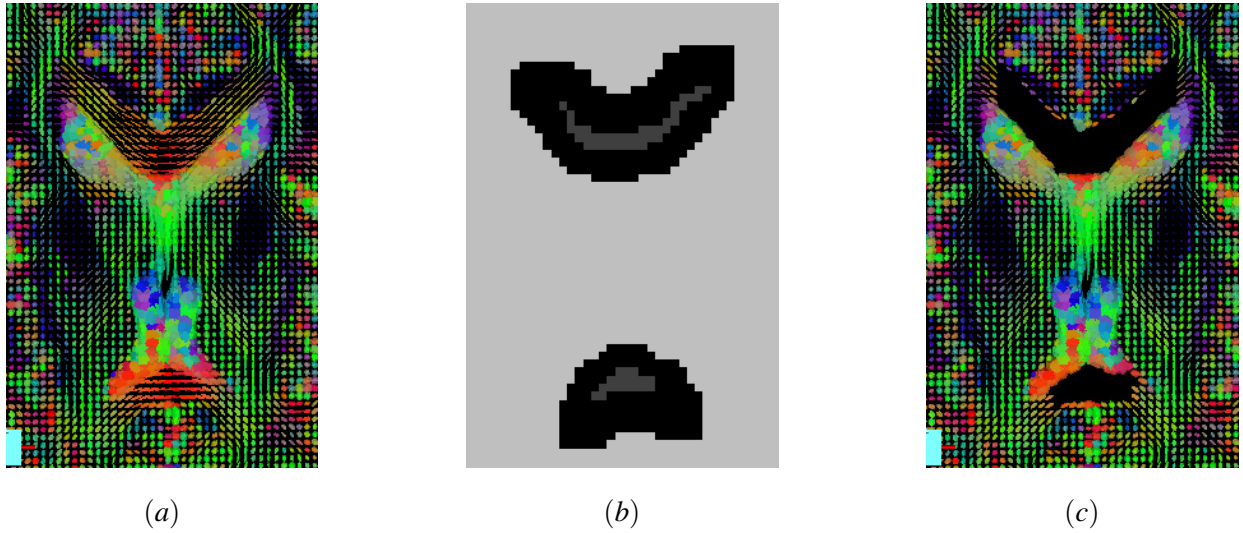


Figure 1.8: Diffusion tensor images segmentation. (a): A close-up on a cross-section of a 3D brain DTI. (b): Image representation of the markers (same cross-section as (a)), obtained from a statistical atlas, for the corpus callosum (dark gray) and for its background (light gray) (c): Segmentation of the corpus callosum by an MSF-cut for the markers. The tensors belonging to the component of the MSF which extends the marker labelled “corpus callosum” are removed from the initial DTI: thus the corresponding voxels appear black.

of the mean curvature of the surface (see [50] for more details). Then, we can compute a watershed cut (in bold in Fig. 1.7b) which leads to a natural and accurate mesh segmentation in the sense that the “borders” of the regions are made of sides of triangles (in bold in Fig. 1.7c) of high curvature.

The direct application of this method on the mesh shown Fig. 1.7d leads to a strong over-segmentation (Fig. 1.7e) due to the huge number of local minima. By using the methodology introduced in section 1.5 to remove from w all the minima which have a depth lower than a predefined threshold (here 50) A watershed cut of the filtered w is depicted in Fig. 1.7f.

In the medical context, *Diffusion Tensor Images (DTIs)* [52] provide a unique insight into oriented structures within tissues. A DTI T maps the set of voxels $\mathcal{V} \subseteq \mathbb{Z}^3$ (i.e. \mathcal{V} is a cuboid of \mathbb{Z}^3) into the set of 3×3 tensors (i.e. 3×3 symmetric positive definite matrices). The value $T(v)$ of a DTI T at a voxel $v \in \mathcal{V}$ describes the diffusion of water molecules at v . For instance, the first eigenvector of $T(v)$ (i.e. the one whose associated eigenvalue is maximal) provides the principal direction of water molecules diffusion at point x and its associated eigenvalue gives the magnitude of the diffusion along this direction. Since water molecules highly diffuse along fiber tracts and since the white matter of the brain is mainly composed of fiber tracts, DTIs are particularly adapted to the study of brain architecture. Fig. 1.8a shows a representation of a cross-section of a brain DTI where the tensors are represented by ellipsoids. Indeed, the data of a tensor is equivalent to the one of an ellipsoid. In the brain, the corpus callosum is an important structure made of

fiber tracts connecting homologous areas of each hemisphere. In order to track the fibers that pass through the corpus callosum, it is necessary to segment it first. The next paragraph briefly reviews how to reach this goal thanks to watershed cuts [47].

We consider the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ induced by the 6-adjacency and defined by $e_{ij} \in \mathcal{E}$ iff $v_i \in \mathcal{V}, v_j \in \mathcal{V}$ and $\sum_{k \in \{1,2,3\}} |v_i^k - v_j^k| = 1$, where $v_i = (v_i^1, v_i^2, v_i^3)$ and $v_j = (v_j^1, v_j^2, v_j^3)$. In order to weight any edge e_{ij} of \mathcal{G} by a dissimilarity measure between the tensors $T(v_i)$ and $T(v_j)$, we choose the Log-Euclidean distance which is known to satisfy interesting properties [53]. Then, we associate to each edge $e_{ij} \in E$ the weight $w_{ij} = \|\log(T(v_i)) - \log(T(v_j))\|$, where \log denotes the matrix logarithm and $\|\cdot\|$ the Euclidean (sometimes also called Frobenius) norm on matrices. To segment the corpus callosum in this graph, we extract (thanks to a statistical atlas), markers for both the corpus callosum and its background and we compute an MSF-cut for these markers. An illustration of this procedure is shown in Figs. 1.8.

To conclude this section, let us mention that spatio-temporal graphs are also feasible. For example, in [54], a 3D+t segmentation of the left-ventricle on cine-MR images was computed, showing an improvement with respect to the segmentation of each 3D volume separately.

1.7 MSF cut hierarchy and saliency maps

We now study some optimality properties of hierarchical segmentations (see [55, 56, 57, 58, 59, 60] for examples of hierarchical segmentations) in the framework of edge-weighted graphs, where the cost of an edge is given by a dissimilarity between two points of an image. Since the pioneering work of [39, 40] stating an equivalence between hierarchies and minimum spanning trees (MST), a large number of hierarchical schemes rely on the construction of such a tree (see [21] for one of the oldest). We formalize a fundamental operation called uprooting that allows us to merge a marked region with one of its neighbors with the cheapest cost. When applied sequentially on the weighted graph of neighboring regions, the uprooting builds a MST of this neighboring graph. Intuitively, one can see that, if one starts from a minimum spanning forest (MSF) rooted in the minima of the image (or, equivalently, from a watershed cut), then one builds a hierarchy of MSFs of the original image itself, the last uprooting step giving an MST of this original image. More surprisingly, Th. 23 states that the two processes are equivalent: any MST of the original image can be built from an uprooting sequence on a watershed cut. Thus, watershed cuts are the only watershed that allows the building of hierarchical segmentations that are optimal with respect to the original image, in the sense that they “preserve” the MST of the original image.

Definition 20 (MSF hierarchy, [14])

Let $\mathcal{M} = \langle M_1, \dots, M_\ell \rangle$ be a sequence of pairwise distinct minima of w and let $\mathcal{T} = \langle \mathcal{G}_0, \dots, \mathcal{G}_\ell \rangle$ be a sequence of subgraphs of \mathcal{G} . We say that \mathcal{T} is an MSF hierarchy for \mathcal{S} if for any $i \in [0, \ell]$, the graph \mathcal{G}_i is an MSF

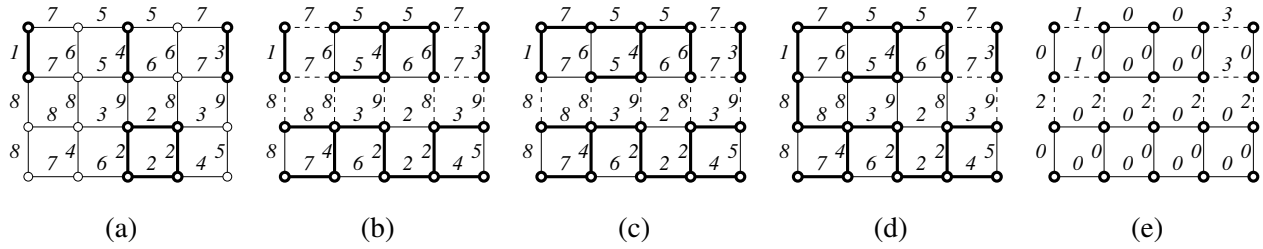


Figure 1.9: (a): An edge-weighted graph \mathcal{G} , the minima of w are depicted in bold. (b): A watershed cut represented by dashed edges; the watershed cut is equivalent to the graph $\mathcal{G}_0 \in MSF(\mathcal{M})$ represented in bold. (c,d): two bold graphs called respectively \mathcal{G}_1 and \mathcal{G}_2 such that $\mathcal{T} = \langle \mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2 \rangle$ is both an MSF hierarchy for and an uprooting by $\langle M_1, M_2 \rangle$ (where M_i is the minimum of w whose altitude is i); their associated cuts are represented by dashed edges. (e): The saliency map of the MSF hierarchy $\langle \mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2 \rangle$.

rooted in $\sqcup[\mathcal{M} \setminus \{M_j \mid j \in [1, i]\}]$, and for any $i \in [1, \ell]$, we have $\mathcal{G}_{i-1} \subseteq \mathcal{G}_i$.

Theorem 21 ([61, 14])

Any \mathcal{G}_i of an MSF hierarchy is a watershed cut of the geodesic reconstruction w_R where the markers are the minima rooting the MSF \mathcal{G}_i .

1.7.1 Uprootings and MSF hierarchies

In this section, we introduce a simple transformation, called uprooting, that allows a forest \mathcal{G}_1 rooted in a graph \mathcal{G}_2 to be incrementally transformed into a forest \mathcal{G}_3 rooted in a graph \mathcal{G}_4 obtained by removing some components of \mathcal{G}_2 . Through an equivalence theorem, we establish an important link between the uprooting transform and the MSF hierarchies. This result opens the way toward efficient algorithms for computing MSF hierarchies.

Let \mathcal{G}_1 be a subgraph of \mathcal{G} that is spanning for \mathcal{V} , and let $v \in \mathcal{V}$. We denote by $CC_v(\mathcal{G}_1)$ the component of \mathcal{G}_1 whose vertex set contains v . Let $\mathcal{V}' \subseteq \mathcal{V}$, we set $CC_{\mathcal{V}'}(\mathcal{G}_1) = \sqcup\{CC_v(\mathcal{G}_1) \mid v \in \mathcal{V}'\}$.

Let $\mathcal{G}_1 \subseteq \mathcal{G}$, and let $e_{ij} = \{v_i, v_j\} \in \mathcal{E}$. The edge e_{ij} is *outgoing from* \mathcal{G}_1 if e_{ij} is made of a vertex in $\mathcal{V}(\mathcal{G}_1)$ and of a vertex in $\overline{\mathcal{V}(\mathcal{G}_1)}$. In the following, by abuse of notation, we write $\mathcal{G}_1 \sqcup \{e_{ij}\}$ for the supremum of \mathcal{G}_1 and the graph induced by $\{e_{ij}\}$: $\mathcal{G}_1 \sqcup \{e_{ij}\} = (\mathcal{V}(\mathcal{G}_1) \cup e_{ij}, \mathcal{E}(\mathcal{G}_1) \cup \{e_{ij}\})$.

Let $\mathcal{G}_1, \mathcal{G}_2$, and \mathcal{G}_3 be three subgraphs of \mathcal{G} such that \mathcal{G}_1 is spanning for \mathcal{V} and such that $\mathcal{G}_1 \neq \mathcal{G}_2$. We say that \mathcal{G}_2 is an *elementary uprooting of* \mathcal{G}_1 by \mathcal{G}_3 if there exists an edge e of minimum weight among the edges outgoing from $CC_{\mathcal{V}(\mathcal{G}_3)}(\mathcal{G}_1)$ such that $\mathcal{G}_2 = \mathcal{G}_1 \sqcup \{u\}$. We also say that \mathcal{G}_2 is an *elementary uprooting of* \mathcal{G}_1 by \mathcal{G}_3 if $\mathcal{G}_2 = \mathcal{G}_1$ and if there is no edge outgoing from $CC_{\mathcal{V}(\mathcal{G}_3)}(\mathcal{G}_1)$.

Definition 22 ([14])

Let $\mathcal{S} = \langle M_1, \dots, M_\ell \rangle$ be a sequence of pairwise distinct minima of w . An uprooting by \mathcal{S} is a sequence

$\langle \mathcal{G}_0, \dots, \mathcal{G}_\ell \rangle$ of graphs such that $\mathcal{G}_0 \in \text{MSF}(\mathcal{M})$ and, for any $i \in [1, \ell]$, \mathcal{G}_i is an elementary uprooting of \mathcal{G}_{i-1} by M_i .

Theorem 23 ([14])

Let $\mathcal{S} = \langle M_1, \dots, M_\ell \rangle$ be a sequence of pairwise distinct minima of w . Let $\mathcal{T} = \langle \mathcal{G}_0, \dots, \mathcal{G}_\ell \rangle$ be a sequence of subgraphs of \mathcal{G} . The sequence \mathcal{T} is an MSF hierarchy for \mathcal{S} if and only if the sequence \mathcal{T} is an uprooting by \mathcal{S} .

1.7.2 Saliency maps

The cuts of the sequence of any MSF hierarchy can be stacked to form a weighting called the saliency map. The saliency map allows us to easily assess the quality and the robustness of a hierarchical segmentation. Furthermore, it is a weighed graph, and can be further processed if needed, for example to remove small regions unwanted in the hierarchy.

Let us give a precise definition of the saliency map [56, 14]. We first need to define, for a graph \mathcal{G} , the map ϕ as $\phi(\mathcal{G}) = \sqcup \{\alpha_2(\mathcal{G}_i) \mid \mathcal{G}_i \text{ is a component of } \mathcal{G}\}$. In other words, $\phi(\mathcal{G})$ is the union of all the graphs induced by the connected components of \mathcal{G} . It is easy to see that ϕ is an opening.

Let $\mathcal{M} = \langle M_1, \dots, M_\ell \rangle$ be a sequence of pairwise distinct minima of w and Let $\mathcal{T} = \langle \mathcal{G}_0, \dots, \mathcal{G}_\ell \rangle$ be a MSF hierarchy for \mathcal{M} . The *saliency map* s for \mathcal{T} is the map such that for any $e_{ij} \in \mathcal{E}$, either $s_{ij} = k$ if there exists k such that k is the lowest number satisfying $e_{ij} \in \phi(\mathcal{G}_k)$ or $s_{ij} = l + 1$ if it does not exists any \mathcal{G}_k such that $e_{ij} \in \phi(\mathcal{G}_k)$. Observe in particular that if $e_{ij} \in \mathcal{E}(\phi(\mathcal{G}_0))$, $s_{ij} = 0$ and that $s[\lambda] = \phi(\mathcal{G}_\lambda)$ for $\lambda \in \{0, \dots, l\}$.

Under the term of *ultrametric watershed*, one can give a computable definition to saliency maps, allowing to show the equivalence between the set of hierarchical segmentations and the set of saliency maps [60].

1.7.3 Application to image segmentation

To use the framework in practice, we have to design an order on the minima of w . Let μ be an attribute on $\mathcal{C}_\mathcal{E}^\leq$ (i.e. a function from $\mathcal{C}_\mathcal{E}^\leq$ to \mathbb{R} that is increasing on $\mathcal{C}_\mathcal{E}^\leq$) such as the area, the volume or the height. We compute an *extinction measure* [62, 63] μ_e to each minima of w by first defining a strict total order relation \prec on the set of minima (based for example on the altitude of each minimum) such that $M_0 \prec M_1 \prec \dots \prec M_l$; then we set $\mu_e(M_0) = \infty$ and

$$\mu_e(M_i) = \min\{\mu([\lambda, C]) \mid [\lambda, C] \in \mathcal{C}_\mathcal{E}^\leq, \text{ there exists } M_j \text{ a minimum of } w, M_j \prec M_i\} \quad (1.5)$$

The map μ_e will define the order \prec_e of the sequence for the hierarchy of MSF: $M_i \prec_e M_j$ whenever $\mu_e(M_i) < \mu_e(M_j)$.

Other choices for the ordering are possible: for example waterfall [64], that consists in computing a sequence of watersheds of watersheds, each step leading to a drastic reduction in the number of basins. Another interesting ordering is done by optimization [57]). Let us consider a two-term-based energy function of the form $\lambda C + D$, where D is a goodness-of-fit term and C is a regularization term. Finding an optimum of this function is NP-hard in the general case. On the other hand, on a hierarchy (hence, on a watershed cut), when the goodness-of-fit term decreases with the fineness of the partitions, and, inversely, that the regularization term increases with this fineness, we can show that finding an optimum can be done in linear time, by dynamic programming. Such an optimization is an efficient way to control the flooding, that is stopped when the optimum is reached. Varying the λ parameter allows us to obtain a complete hierarchy of segmentations.

Fig. 1.10b,c, and d illustrates some saliency maps on the image of Fig. 1.10a. The underlying graph is the one induced by the 4-adjacency relation whose edges are weighted by a simple color gradient (maximum, over the RGB channels, of the absolute differences of pixel values). The minima are ordered thanks to extinction values [62] related to depth, volume and color consistency.

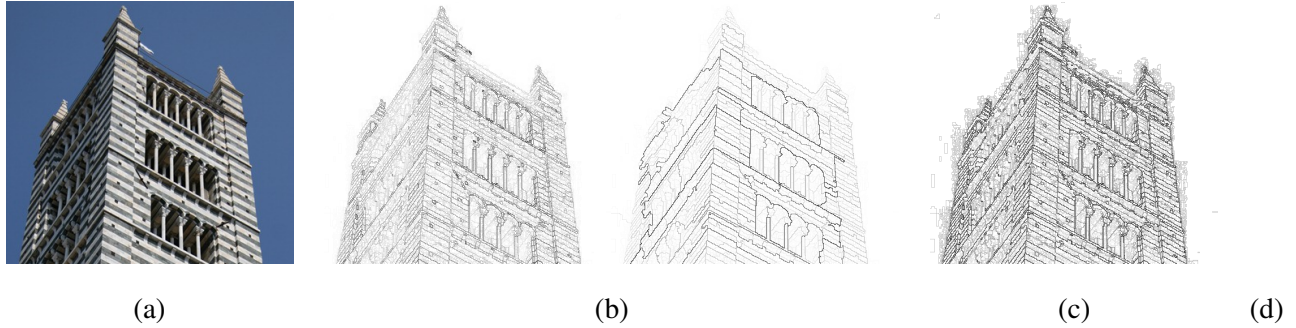


Figure 1.10: Illustration of saliencies of watershed cuts (original picture (a) from *koakoo*: <http://blog.photos-libres.fr/>).

1.8 Optimization and the Power Watershed

In this section, we review the Power Watershed framework [15] (see also Leo Grady's chapter, this book). In the previous sections dealing with watershed cuts, the weights encode a dissimilarity such as a gradient. Classically, in the context of segmentation and clustering applications based on optimization, the weights encode affinity such that vertices connected by an edge with high weight are considered to be strongly connected and edges with a low weight represent nearly disconnected vertices. One common choice for generating weights from image intensities is to set a Gaussian weighting such as

$$w_{ij} = \exp(-\beta(\nabla I)^2), \quad (1.6)$$

$\begin{array}{c} p \\ \backslash \\ q \end{array}$	0	finite	∞
1	Collapse to seeds	Max-Flow (Graph cuts)	Power watershed $q = 1$
2	ℓ_2 norm Voronoi	Random walker	Power watershed $q = 2$
∞	ℓ_1 norm Voronoi	ℓ_1 norm Voronoi	Shortest Path Forest

Table 1.1: The generalized scheme of [15] for image segmentation includes several popular segmentation algorithms as special cases of the parameters p and q . The power watershed may be optimized efficiently with a maximum spanning forest calculation.

where ∇I is the normalized gradient of the image I . The gradient for a grey level image is $I_i - I_j$. As the weights are inverted, the maxima are considered instead of minima, and a thalweg is computed instead of watershed. A thalweg is the deepest continuous line along a valley. In the rest of the paper, we continue to use by convention the term “watershed” instead of “thalweg”.

Given foreground F and background B seeds, and p, q two real positive values, the energy presented for binary segmentation in [15] is given by

$$x^{*pq} = \arg \min_x \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q \quad \text{s.t. } x(F) = 1, x(B) = 0. \quad (1.7)$$

In this energy, w_{ij} can be interpreted as a weight enforcing a regularization of the contours, such that any (usually unwanted) high-frequency content is penalized in x . The energy defined in (1.7) essentially forces x to remain smooth within the object, while allowing it to vary quickly close to point clusters near the boundary of the object. The data constraints enforce fidelity of x to a specified configuration, taking the values zero and one as the reconstructed object indicator. Observe that the values of x may not necessarily be binary when the value of q is strictly greater than one.

The different values of p and q lead to different algorithms for optimizing the energy. Those algorithms form the underpinning for many of the advanced image segmentation methods in the literature. Table 1.1 gives a reference for the different algorithms generated by various values of p and q . The limit cases are the limit of the minimizers of eq. (1.7), *e.g.*, the case $p \rightarrow \infty$ reads

$$x^{*q} = \lim_{p \rightarrow \infty} x^{*pq}. \quad (1.8)$$

Let us highlight the main choices for those parameters.

- When the power on the weight, p , is finite, and the exponent $q = 1$, we recover [61] the Max-Flow (Graph Cuts) energy which can be optimized by a max flow algorithm.

- When $q = 2$, we obtain a combinatorial Dirichlet problem also known as the Random walker problem [65].
- When q and p converge toward infinity with the same speed, then [66] a solution to (1.8) can be computed by the shortest path (geodesics) algorithm [6, 67].
- As described in [15], by raising the power p toward infinity and varying the power q we obtain a family of segmentation models which we refer to as **power watershed**, that we detail below.

A primary advantage of power watershed with varying q is that the main computational burden of these algorithms depends on an MSF computation, which is extremely efficient [68]. For example, interpreted from the standpoint of the Gaussian weighting function in (1.6), it is clear that we may associate $\beta = p$ to understand that the watershed equivalence comes from operating the weighting function in a particular parameter range. In the case $q = 1$, an important insight from this connection is that *above some value of β we can replace the expensive max-flow computation with an efficient maximum spanning forest computation.*

Let us review some important theoretical results. First, we highlight that the cut obtained when minimizing the energy (1.8) is a watershed cut [13], and thus a Maximum Spanning Forests [13] (MSF). Let x^* be a solution to (1.8) and s be the segmentation result defined as $s_i = 1$ if $x_i^* \geq \frac{1}{2}$, 0 if $x_i^* < \frac{1}{2}$. The set of edges e_{ij} such that $s_i \neq s_j$ is a q -cut.

Theorem 24 ([15])

For $q \geq 1$, any q -cut is a watershed cut of the (geodesically) reconstructed weights.

Furthermore, if every connected component of \mathcal{M} contains at least a vertex of $B \cup F$, and $q \geq 1$, then any q -cut when $p \rightarrow \infty$ is an MSF cut for w .

An algorithm is presented in Algorithm 1 to optimize the (unique) watershed that optimizes the energy for $q \geq 1$ and $p \rightarrow \infty$. The set of parameters $p \rightarrow \infty$ and $q = 2$ is particularly interesting.

1. The power watershed algorithm in the case $q = 2$ has a well-defined behavior in the absence or lack of weight information (presence of plateaus); indeed, as the energy is strictly convex, the solution to (1.8) is unique.
2. The worst-case complexity of the power watershed algorithm in the case $p \rightarrow \infty$ is given by the cost of optimizing (1.7) for the given q . In the best-case scenario (all weights have unique values), the power watershed algorithm has the same asymptotic complexity as the algorithm used for a MSF computation (quasi-linear). In practical applications where the plateaus have a size less than some fixed value K , then the complexity of the power watershed algorithm matches the quasi-linear complexity of the standard watershed-cut algorithm.

Algorithm 1: Power watersheds algorithm $p \rightarrow \infty, q > 1$

Data: A weighted graph $G(V, E)$ comprising known labels $x(B), x(F)$.**Result:** A **potential function** \bar{x} (solution of (1.8) thanks to Th. 25.)**while** *any node has an unknown label* **do** Find a maximal subgraph $S \in G$ composed of edges of maximal weight; **if** *S contains any nodes with known x* **then** Find x_S minimizing (1.7) for q on the subset S ; Consider all x_S values produced by this operation as known; **else** Merge all of the nodes in S into a single node, such that when the value of x for this merged node becomes known, all merged nodes are assigned the same value of x and considered known;

The main properties of Algorithm 1 (with $q \geq 1$) are summarized in the following theorem. This theorem states that the energy found by the algorithm is the correct one, *i.e.* $x^{*q} = \bar{x}$, and that furthermore, the computed cut is a MSF cut.

Theorem 25 ([15])

If $q > 1$, the potential x^ obtained by minimizing the energy of (1.7) subject to the boundary constraints converges toward the potential \bar{x} obtained by Algorithm 1 as $p \rightarrow \infty$.*

Furthermore, for any $q \geq 1$, the cut C defined by the segmentation s computed by Algorithm 1 is an MSF cut for w .

1.8.1 Applications

Minimizing exactly the energy $E_{1,1}$ is possible by using the graph cuts (*i.e.* $q = 1$) algorithm in the case of two labels, but is NP-hard if constraints impose more than two different labels. However, the other algorithms presented in the framework (and in particular the watershed cuts and the power watershed) can efficiently perform seeded segmentation with as many labels as desired [15]. As several examples of segmentation have already been shown in this chapter, we rather highlight two other applications that take advantage the unique optimization characteristics of the power watershed.

The first example is *shape fitting*: surface reconstruction from a set of noisy point measurements has been a well studied problem over the last decades. Recently, variational [71, 72] and discrete optimization [73] approaches have been applied to solve it, demonstrating good robustness to outliers thanks to a global energy minimization scheme. In [69], Couprie *et al.* use the power watershed framework to derive a specific

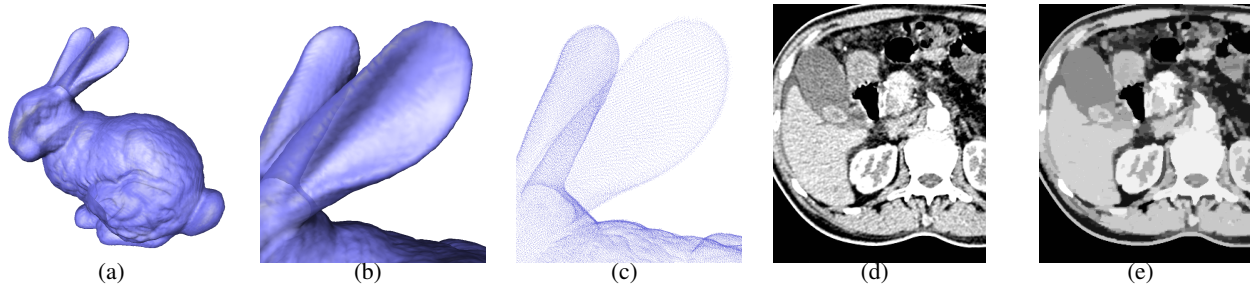


Figure 1.11: Two applications of power watershed. (a-c) Shape fitting using Power Watershed [69]. (a) Reconstructed bunny. (b) A close-up on one of the ears. (c) A close-up on the set of original scanned noisy points measurements. (d-e) Filtering of a liver image by anisotropic diffusion driven by power watershed [70]. In (e) noise and small vessels are both removed from (d), leading to a result which may be used as a first step before segmentation.

watershed algorithm for surface reconstruction. The proposed algorithm is fast, robust to marker placement, and produces smooth surfaces. Figure 1.11(a-c) shows a surface reconstructed from noisy scanned dot sets using the power watershed algorithm.

The second example deals with filtering. In [70], Couprie *et al.* reformulate the problem of anisotropic diffusion as an L_0 optimization problem, and show that power watersheds are able to optimize this energy quickly and effectively. This study paves the way for using the power watershed as a useful general-purpose minimizer in many different computer vision contexts. An example of such an L_0 optimization is presented in Fig. 1.11(d-e).

1.9 Conclusion

We have presented in this paper some applications of mathematical morphology to weighted graphs. The translation of the abstract framework of lattices to graphs allows us to obtain morphological operators thinner (in the sense that they process smaller details) than the usual ones usually defined only on the vertices. We have shown how to filter an image using connected filters based on a tree representation of the image. We have exhibited the links between watersheds and minimum spanning trees, allowing us to make the equivalence between hierarchical segmentation based on watershed cuts and minimum spanning trees. We conclude the chapter by exploring links between the morphological and the optimization approaches.

For practical applications, we want to stress the importance of having a framework with an open-source generic implementation of existing algorithms, not limited to the pixel framework, but also able to deal transparently with edges, or, more generally, with graphs and complexes [74].

Bibliography

- [1] G. Matheron, *Random sets and integral geometry*. New York: John Wiley & Sons, 1975.
- [2] J. Serra, *Image analysis and mathematical morphology*. London, U.K.: Academic Press, 1982.
- [3] J. Serra, Ed., *Image analysis and mathematical morphology. Volume 2: Theoretical advances*. London, U.K.: Academic Press, 1988.
- [4] L. Vincent, “Graphs and mathematical morphology,” *Signal Processing*, vol. 16, pp. 365–388, 1989.
- [5] H. Heijmans and L. Vincent, “Graph morphology in image analysis,” in *Mathematical morphology in image processing*, E. Dougherty, Ed. New York: Marcel Dekker, 1993, vol. 34, ch. 6, pp. 171–203.
- [6] A. X. Falcão, R. A. Lotufo, and G. Araujo, “The image foresting transformation,” *IEEE PAMI*, vol. 26, no. 1, pp. 19–29, 2004.
- [7] F. Meyer and R. Lerallut, “Morphological operators for flooding, leveling and filtering images using graphs,” in *Graph-based Representations in Pattern Recognition (GbrPR’07)*, vol. LNCS 4538, 2007, pp. 158–167.
- [8] F. Meyer and J. Angulo, “Micro-viscous morphological operators,” in *Mathematical Morphology and its Application to Signal and Image Processing (ISMM 2007)*, 2007, pp. 165–176.
- [9] V.-T. Ta, A. Elmoataz, and O. Lézoray, “Partial difference equations over graphs: Morphological processing of arbitrary discrete data,” in *Computer Vision - ECCV 2008*, ser. Lecture Notes in Computer Science, D. Forsyth, P. Torr, and A. Zisserman, Eds. Springer Berlin / Heidelberg, 2008, vol. 5304, pp. 668–680.
- [10] J. Cousty, L. Najman, and J. Serra, “Some morphological operators in graph spaces,” in *International Symposium on Mathematical Morphology 2009*, ser. LNCS, vol. 5720. Springer Verlag, Aug. 2009, pp. 149–160.
- [11] I. Bloch and A. Bretto, “Mathematical morphology on hypergraphs: Preliminary definitions and results,” in *Discrete Geometry for Computer Imagery*, ser. Lecture Notes in Computer Science, I. Debled-Rennesson, E. Dmenjoud, B. Kerautret, and P. Even, Eds. Springer Berlin / Heidelberg, 2011, vol. 6607, pp. 429–440.
- [12] F. Dias, J. Cousty, and L. Najman, “Some morphological operators on simplicial complexes spaces,” in *DGCI 2011*, ser. LNCS, no. 6607. Springer, 2011, pp. 441–452.
- [13] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, “Watershed Cuts: Minimum Spanning Forests and the Drop of Water Principle,” *IEEE PAMI*, vol. 31, no. 8, pp. 1362–1374, Aug. 2009.

- [14] J. Cousty and L. Najman, "Incremental algorithm for hierarchical minimum spanning forests and saliency of watershed cuts," in *ISMM 2011*, ser. LNCS, no. 6671, 2011, pp. 272–283.
- [15] C. Couprie, L. Grady, L. Najman, and H. Talbot, "Power Watersheds: A Unifying Graph Based Optimization Framework," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 7, pp. 1384–1399, July 2011, <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2010.200>. [Online]. Available: <http://powerwatershed.sourceforge.net/>
- [16] P. Soille, *Morphological Image Analysis*. Springer-Verlag, 1999.
- [17] L. Najman and H. Talbot, Eds., *Mathematical morphology: from theory to applications*. ISTE-Wiley, June 2010.
- [18] Nesetril, Milkova, and Nesetrilova, "Otakar Boruvka on Minimum Spanning Tree problem: Translation of both the 1926 papers, comments, history," *DMATH: Discrete Mathematics*, vol. 233, 2001.
- [19] G. Choquet, "étude de certains réseaux de routes," *Comptes-rendus de l'Acad. des Sciences*, pp. 310–313, 1938.
- [20] C. Zahn, "Graph-theoretical methods for detecting and describing Gestalt clusters," *IEEE Transactions on Computers*, vol. C-20, no. 1, pp. 99–112, 1971.
- [21] O. J. Morris, M. d. J. Lee, and A. G. Constantinides, "Graph theory for image analysis: an approach based on the shortest spanning tree," *IEE proc. on communications, radar and signal*, vol. 133, no. 2, pp. 146–152, 1986.
- [22] F. Meyer, "Minimum spanning forests for morphological segmentation," in *Procs. of the second international conference on Mathematical Morphology and its Applications to Image Processing*, September 1994, pp. 77–84.
- [23] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, pp. 167–181, 2004.
- [24] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE PAMI*, vol. 13, no. 6, pp. 583–598, June 1991.
- [25] G. Birkhoff, *Lattice Theory*, ser. American Mathematical Society Colloquium Publications. American Mathematical Society, 1995, vol. 25.
- [26] C. Ronse and J. Serra, "Algebraic foundations of morphology," in *Mathematical Morphology*, L. Najman and H. Talbot, Eds. ISTE-Wiley, 2010, pp. 35–79.
- [27] P. Salembier and J. Serra, "Flat zones filtering, connected operators, and filters by reconstruction," *IEEE TIP*, vol. 4, no. 8, pp. 1153–1160, Aug 1995.
- [28] C. Ronse, "Set-theoretical algebraic approaches to connectivity in continuous or digital spaces," *JMIV*, vol. 8, no. 1, pp. 41–58, 1998.
- [29] U. Braga-Neto and J. Goutsias, "Connectivity on complete lattices: new results," *Comput. Vis. Image Underst.*, vol. 85, no. 1, pp. 22–53, 2002.
- [30] G. K. Ouzounis and M. H. Wilkinson, "Mask-based second-generation connectivity and attribute filters," *IEEE PAMI*, vol. 29, no. 6, pp. 990–1004, 2007.

- [31] R. K. Gabriel and R. R. Sokal, "A new statistical approach to geographic variation analysis," *Systematic Zoology*, vol. 18, no. 3, pp. 259–278, Sep. 1969.
- [32] P. Salembier, A. Oliveras, and L. Garrido, "Anti-extensive connected operators for image and sequence processing," *IEEE TIP*, vol. 7, no. 4, pp. 555–570, April 1998.
- [33] L. Najman and M. Couprie, "Building the component tree in quasi-linear time," *IEEE TIP*, vol. 15, no. 11, pp. 3531–3539, 2006.
- [34] P. Salembier and L. Garrido, "Binary partition tree as an efficient representation for image processing, segmentation and information retrieval," *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 561–576, Apr. 2000.
- [35] V. Caselles and P. Monasse, *Geometric Description of Images as Topographic Maps*, ser. Lecture Notes in Computer Science. Springer, 2010, vol. 1984.
- [36] P. Salembier, "Connected operators based on tree pruning strategies," in *Mathematical morphology: from theory to applications*, L. Najman and H. Talbot, Eds. ISTE-Wiley, 2010, pp. 179–198.
- [37] J. Benzécri, *L'Analyse des données: la Taxinomie*. Dunod, 1973, vol. 1.
- [38] B. Leclerc, "Description combinatoire des ultramétries," *Mathématique et sciences humaines*, vol. 73, pp. 5–37, 1981.
- [39] N. Jardine and R. Sibson, *Mathematical taxonomy*. Wiley, 1971.
- [40] J. Gower and G. Ross, "Minimum spanning tree and single linkage cluster analysis," *Appl. Stats.*, vol. 18, pp. 54–64, 1969.
- [41] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical Society*, vol. 7, pp. 48–50, February 1956.
- [42] F. Meyer and S. Beucher, "Morphological segmentation," *JVCIR*, vol. 1, no. 1, pp. 21–46, Sept. 1990.
- [43] S. Beucher and F. Meyer, "The morphological approach to segmentation: the watershed transformation," in *Mathematical morphology in image processing*, ser. Optical Engineering, E. Dougherty, Ed. New York: Marcel Dekker, 1993, vol. 34, ch. 12, pp. 433–481.
- [44] L. Najman and M. Schmitt, "Watershed of a continuous function," *Signal Processing*, vol. 38, no. 1, pp. 99–112, 1994.
- [45] J. B. T. M. Roerdink and A. Meijster, "The watershed transform: Definitions, algorithms and parallelization strategies," *Fundamenta Informaticae*, vol. 41, no. 1-2, pp. 187–228, 2001.
- [46] G. Bertrand, "On topological watersheds," *JMIV*, vol. 22, no. 2-3, pp. 217–230, May 2005.
- [47] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Watershed cuts: thinnings, shortest-path forests and topological watersheds," *IEEE PAMI*, vol. 32, no. 5, pp. 925–939, 2010.
- [48] F. Meyer, "Topographic distance and watershed lines," *Signal Processing*, vol. 38, no. 1, pp. 113–125, July 1994.

- [49] P. Soille and C. Gratin, “An efficient algorithm for drainage networks extraction on DEMs,” *Journal of Visual Communication and Image Representation*, vol. 5, no. 2, pp. 181–189, June 1994.
- [50] S. Philipp-Foliguet, M. Jordan, L. Najman, and J. Cousty, “Artwork 3D Model Database Indexing and Classification,” *Pattern Recogn.*, vol. 44, no. 3, pp. 588–597, Mar. 2011.
- [51] L. J. Grady and J. R. Polimeni, *Discrete Calculus: Applied Analysis on Graphs for Computational Science*, 1st ed. Springer, Aug. 2010.
- [52] P. J. Basser, J. Mattiello, and D. LeBihan, “MR diffusion tensor spectroscopy and imaging,” *Biophys. J.*, vol. 66, no. 1, pp. 259–267, 1994.
- [53] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, “Log-Euclidean metrics for fast and simple calculus on diffusion tensors,” *Magnetic Resonance in Medicine*, vol. 56, no. 2, pp. 411–421, August 2006.
- [54] J. Cousty, L. Najman, M. Couprie, S. Clément-Guinaudeau, T. Goissen, and J. Garot, “Segmentation of 4D cardiac MRI: automated method based on spatio-temporal watershed cuts,” *IVC*, vol. 28, no. 8, pp. 1229–1243, Aug. 2010.
- [55] F. Meyer, “The dynamics of minima and contours,” in *Mathematical Morphology and its Applications to Image and Signal Processing*, P. Maragos, R. Schafer, and M. Butt, Eds. Boston: Kluwer, 1996, pp. 329–336.
- [56] L. Najman and M. Schmitt, “Geodesic saliency of watershed contours and hierarchical segmentation,” *IEEE PAMI*, vol. 18, no. 12, pp. 1163–1173, December 1996.
- [57] L. Guigues, J. P. Cocquerez, and H. L. Men, “Scale-sets image analysis,” *IJCV*, vol. 68, no. 3, pp. 289–317, 2006.
- [58] P. A. Arbeláez and L. D. Cohen, “A metric approach to vector-valued image segmentation,” *IJCV*, vol. 69, no. 1, pp. 119–126, 2006.
- [59] F. Meyer and L. Najman, “Segmentation, minimum spanning tree and hierarchies,” in *Mathematical Morphology: from theory to application*, L. Najman and H. Talbot, Eds. London: ISTE-Wiley, 2010, ch. 9, pp. 229–261.
- [60] L. Najman, “On the equivalence between hierarchical segmentations and ultrametric watersheds,” *Journal of Mathematical Imaging and Vision*, vol. 40, no. 3, pp. 231–247, July 2011, arXiv:1002.1887v2. [Online]. Available: <http://www.laurentnajman.org>
- [61] C. Allène, J.-Y. Audibert, M. Couprie, and R. Keriven, “Some links between extremum spanning forests, watersheds and min-cuts,” *IVC*, vol. 28, no. 10, pp. 1460–1471, Oct. 2010.
- [62] C. Vachier and F. Meyer, “Extinction value: a new measurement of persistence,” in *IEEE Workshop on Nonlinear Signal and Image Processing*, 1995, pp. 254–257.
- [63] G. Bertrand, “On the dynamics,” *IVC*, vol. 25, no. 4, pp. 447–454, 2007.
- [64] S. Beucher, “Watershed, hierarchical segmentation and waterfall algorithm,” in *Mathematical Morphology and its Applications to Image Processing*, J. Serra and P. Soille, Eds. Kluwer Academic Publishers, 1994, pp. 69–76.

- [65] L. Grady, “Random walks for image segmentation,” *IEEE PAMI*, vol. 28, no. 11, pp. 1768–1783, 2006.
- [66] A. K. Sinop and L. Grady, “A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm,” in *Proc. of ICCV’07*, 2007.
- [67] G. Peyre, M. Pechaud, R. Keriven, and L. Cohen, “Geodesic methods in computer vision and graphics,” *Foundations and Trends in Computer Graphics and Vision*, vol. 5, no. 3-4, pp. 197–397, 2010. [Online]. Available: <http://hal.archives-ouvertes.fr/hal-00528999/>
- [68] B. Chazelle, “A minimum spanning tree algorithm with inverse-Ackermann type complexity,” *Journal of the ACM*, vol. 47, pp. 1028–1047, 2000.
- [69] C. Couprie, X. Bresson, L. Najman, H. Talbot, and L. Grady, “Surface reconstruction using power watershed,” in *ISMM 2011*, ser. LNCS, no. 6671, 2011, pp. 381–392.
- [70] C. Couprie, L. Grady, L. Najman, and H. Talbot, “Anisotropic Diffusion Using Power Watersheds,” in *International Conference on Image Processing (ICIP’10)*, Sept. 2010, pp. 4153–4156.
- [71] T. Goldstein, X. Bresson, and S. Osher, “Geometric applications of the Split Bregman Method: Segmentation and surface reconstruction,” UCLA, Computational and Applied Mathematics Reports, Tech. Rep. 09-06, 2009.
- [72] J. Ye, X. Bresson, T. Goldstein, and S. Osher, “A fast variational method for surface reconstruction from sets of scattered points,” UCLA, Computational and Applied Mathematics Reports, Tech. Rep. 10-01, 2010.
- [73] V. Lempitsky and Y. Boykov, “Global Optimization for Shape Fitting,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, USA, 2007.
- [74] R. Levillain, T. Géraud, and L. Najman, “Why and How to Design a Generic and Efficient Image Processing Framework: The Case of the Milena Library,” in *17th International Conference on Image Processing*, 2010, pp. 1941–1944.

Index

- adjunction, 6
- anisotropic diffusion, 26
- catchment basin, 15
- closing, 8
- component tree, 12
- dilation, 6
- duality, 7
- edge, 3
- erosion, 6
- extension, 14
- extinction, 21
- filter, 8
- flooding, 12
- forest
 - minimum spanning, 16
- graph, 3
 - induced, 14
- graph cuts, 23
- hierarchy, 13
 - MSF, 19
- infimum, 3
- lattice, 3
 - complete, 3
- mesh, 17
- minimum spanning tree, 16
- MSF, *see* forest, minimum spanning
- opening, 8
- point, 3
- random walker, 24
- reconstruction
 - geodesic, 13
- saliency map, 21
- shape fitting, 25
- supremum, 3
- tree
 - component, 12
- uprooting, 20
- vertex, 3
- watershed
 - cut, 14
 - power, 24
 - ultrametric, 21