Watersheds, minimum spanning forests, and the drop of water principle

J. Cousty, G. Bertrand, L. Najman, and M. Couprie

Institut Gaspard-Monge

Laboratoire A2SI, Groupe ESIEE

Cité Descartes, BP99, 93162 Noisy-le-Grand Cedex France

Abstract

In this paper, we study the watersheds in edge-weighted graphs. Contrarily to previous work, we define the watersheds following the intuitive idea of drops of water flowing on a topographic surface. We establish the consistency of these watersheds and proved their optimality in terms of minimum spanning forests. We introduce a new local transformation on maps which equivalently define these watersheds and derive two linear-time algorithms. To our best knowledge, similar properties are not verified in other frameworks and the two proposed algorithms are the most efficient existing algorithms, both in theory and practice. Afterward, we investigate the mathematical links and differences with two other segmentation methods, *i.e.*, the Image Foresting Transform and the topological watershed. Finally, the defined concepts are illustrated in image segmentation leading to the conclusion that the proposed approach improves the quality of watershed-based segmentations.

Key words: Watershed, minimum spanning forest, minimum spanning tree, graph, mathematical morphology, image segmentation

---

Introduction

The watershed has been extensively studied during the 19th century by Maxwell [1] and Jordan [2] among others. One hundred years later, the watershed transform was introduced by Beucher and Lantuéjoul [3] for image segmentation and is now used as a fundamental step in many powerful segmentation procedures.

Email addresses: j.cousty@esiee.fr (J. Cousty), g.bertrand@esiee.fr (G. Bertrand), l.najman@esiee.fr (L. Najman), m.couprie@esiee.fr (M. Couprie).

Let us consider a grayscale image as a topographic surface: the gray level of a pixel becomes the elevation of a point, the basins and valleys of the topographic surface correspond to dark areas, whereas the mountains and crest lines correspond to the light areas. The watershed divide may be thought of as a separating set of points along which a drop of water can flow down towards at least two regional minima.

In order to compute the watershed of a digital image, several approaches have been proposed and many of them consider a grayscale digital image as a vertex-weighted graph. One of the most popular consists of simulating a flooding of the topographic surface from its regional minima [4–6]. The divide is made of "dams" built at those points where water coming from different minima would meet. Another approach, called topological watershed [7–9], allows to rigorously define the notion of a watershed in a discrete space and to prove important properties not guaranteed by most watershed algorithms [10]. It consists of lowering the values of a map (*e.g.*, the grayscale image) while preserving some topological properties, namely, the number of connected components of each lower cross-section. In this case, the watershed divide is the set of points not in any regional minimum of the transformed map.

In this paper, we investigate a different framework: we consider a graph whose edges are weighted by a cost function (see, for example, [11] and [12, 13]).

We propose a new definition of watershed, called watershed cut, and obtain a set of remarkable properties. Unlike previous works, watersheds cuts are defined following the intuitive idea of drops of water flowing on a topographic surface.

Our first contribution establishes the consistency of watershed cuts. In particular, we prove that they 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, our definition is the first one that satisfies such a property.

Our second contribution establishes the optimality of watershed cuts. In [11], F. Meyer shows the link between minimum spanning forests (MSF) and flooding from marker algorithms. In this paper, we extend the problem of minimum spanning forests and show the equivalence between watershed cuts and separations induced by minimum spanning forest relative to the minima.

Our third contribution consists of a new thinning paradigm to compute watersheds in linear time. More precisely, we propose a new transformation, called border thinning, that lowers the values of edges that match a simple local configuration until idempotence. The minima of the transformed map constitute a minimum spanning forest relative to the minima of the original one and, hence, induce a watershed cut. Moreover, any such minimum spanning forest can be obtained by this transformation. We discuss the possibility of parallel algorithms based on this transformation and give a sequential implementation which runs in linear time.

Our fourth contribution is a second linear-time algorithm that does not need minima extraction. Contrarily to previously published algorithms, the two algorithms proposed in this paper do not require any sorting step, nor the use of any hierarchical queue. Therefore, they both run in linear time whatever the range of the input map. To our best knowledge, these are the first watershed algorithms with such properties.

Our fifth contribution consists of a mathematical comparison between watershed cuts and two other segmentation paradigms. We first study the links and differences with shortest-path forests (the theoretical basis of the Image Foresting Transform [12] and the fuzzy connected image segmentation [14,15]). We show that any minimum spanning forest is a shortest-path forest and that the converse is, in general, not true. Then, we investigate the topological watershed. We prove that any border thinning is a W-thinning (*i.e.*, the transformation which allows to obtain a topological watershed). A major consequence is that border thinnings, and, thus, watershed cuts inherit the mathematical properties of topological watersheds [8, 10].

Finally, we illustrate that the proposed watershed localizes with better accuracy the contours of objects in digital images. To this aim, we provide, on few examples, results of morphological schemes based on watersheds in vertex weighted graphs and results of their adaptation in edge-weighted graphs.

1 Basic notions and notations

This paper is settled in the framework of edge-weighted graphs. Following the notations of [16], we present some basic definitions to handle such kind of graphs.

1.1 Graphs

We define a *graph* as a pair $X = (V(X), E(X))$ where $V(X)$ is a finite set and $E(X)$ is composed of unordered pairs of $V(X)$, *i.e.*, $E(X)$ is a subset of $\{\{x, y\} \subseteq V(X) \mid x \neq y\}$. Each element of $V(X)$ is called a *vertex or a point (of X)*, and each element of $E(X)$ is called an *edge (of X)*. If $V(X) \neq \emptyset$, we say that X is *non-empty.*
Let X be a graph. If $u = \{x, y\}$ is an edge of X, we say that x and y are

adjacent (for X). Let $\pi = \langle x_0, \ldots, x_l \rangle$ be an ordered sequence of vertices of X, π is *a path from x_0 to x_l in X (or in $V(X)$)* if for any $i \in [1, l]$, x_i is adjacent to x_{i-1}. In this case, we say that *x_0 and x_l are linked for X*. If $l = 0$, then *π is a trivial path in X*. We say that *X is connected* if any two vertices of X are linked for X.

Let X and Y be two graphs. If $V(Y) \subseteq V(X)$ and $E(Y) \subseteq E(X)$, we say that *Y is a subgraph of X* and we write $Y \subseteq X$. We say that Y is a *connected component of X*, or simply a *component of X*, if Y is a connected subgraph of X which is maximal for this property, *i.e.*, for any connected graph Z, $Y \subseteq Z \subseteq X$ implies $Z = Y$.

Throughout this paper G denotes a connected graph. In order to simplify the notations, this graph will be denoted by $G = (V, E)$ instead of $G = (V(G), E(G))$. We will also assume that $E \neq \emptyset$.

In applications to image segmentation, V is the set of picture elements (pixels) and E is any of the usual adjacency relations, *e.g.*, the 4- or 8-adjacency in 2D [17].

Let $X \subseteq G$. An edge $\{x, y\}$ of G is *adjacent to X* if $\{x, y\} \cap V(X) \neq \emptyset$ and if $\{x, y\}$ does not belong to $E(X)$; in this case and if y does not belong to $E(X)$, we say that $\{x, y\}$ is *outgoing from X* and that *y is adjacent to X*. If π is a path from x to y and y is a vertex of X, then π is a *path from x to X (in G)*.

If S is a subset of E, we denote by *$\overline{S}$ the complementary set of S in E*, *i.e.*, $\overline{S} = E \setminus S$.
Let $S \subseteq E$, *the graph induced by S* is the graph whose edge set is S and whose vertex set is made of all points which belong to an edge in S, *i.e.*, $(\{x \in V \mid \exists u \in S, x \in u\}, S)$. In the following, when no confusion may occur, the graph induced by S is also denoted by S.

1.2 Edge-weighted graphs

We denote by $\mathcal{F}$ the set of all maps from E to $\mathbb{Z}$.

Let $F \in \mathcal{F}$. If u is an edge of G, $F(u)$ is the *altitude* of u. Let $X \subseteq G$ and $k \in \mathbb{Z}$. A subgraph X of G is a *minimum of F (at altitude k)* if:
- X is connected; and
- k is the altitude of any edge of X; and
- the altitude of any edge adjacent to X is strictly greater than k.

We denote by $M(F)$ the graph whose vertex set and edge set are, respectively, the union of the vertex sets and edge sets of all minima of F.

In the sequel of this paper, F denotes an element of $\mathcal{F}$.

For applications to image segmentation, we will assume that the altitude of u, an edge between two pixels x and y, represents the dissimilarity between x and y (*e.g.*, $F(u)$ equals the absolute difference of intensity between x and y). Thus, we suppose that the salient contours are located on the highest edges of G.

2 Watersheds

The intuitive idea underlying the notion of a watershed comes from the field of topography: a drop of water falling on a topographic surface follows a descending path and eventually reaches a minimum. The watershed may be thought of as the separating lines of the domain of attraction of drops of water. Despite its simplicity, none of the classical definitions formalize this intuitive idea. In this paper, contrarily to previous works, we follow the drop of water principle to define the notion of a watershed in an edge-weighted graph.

2.1 Extensions and graph cuts

We present the notions of extension and graph cut which play an important role for defining a watershed in an edge-weighted graph.
Intuitively, the regions of a watershed (also called catchment basins) are associated with the regional minima of the map. Each catchment basin contains a unique regional minimum, and conversely, each regional minimum is included in a unique catchment basin: the regions of the watershed "extend" the minima. In [8], G. Bertrand formalizes the notion of extension.

Definition 1 (from Def. 12 in [8]) *Let X and Y be two non-empty subgraphs of G. We say that Y is an* extension of X (in G) *if $X \subseteq Y$ and if any component of Y contains exactly one component of X.*

The graphs (drawn in bold) in Fig. 1b and c are two extensions of the one depicted in Fig. 1a.

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.

Definition 2 *Let $X \subseteq G$ and $S \subseteq E$. We say that S is a* (graph) cut for X

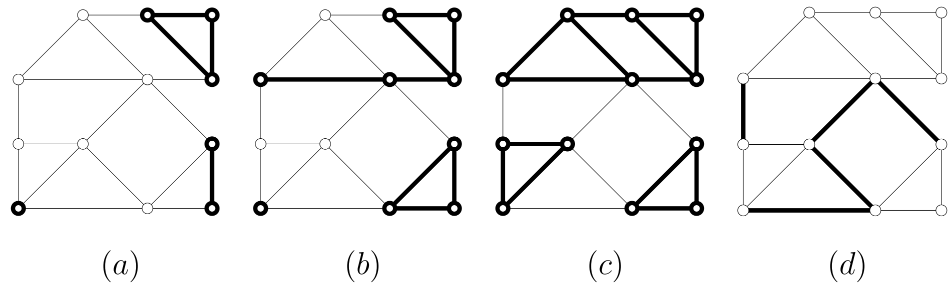


Fig. 1. A graph G. The set of vertices and edges represented in bold is: (a), a subgraph X of G; (b), an extension of X; (c): an extension Y of X which is maximal; and (d): a cut S for X such that $\overline{S} = Y$.

if $\overline{S}$ *is an extension of* X *and if* S *is minimal for this property,* i.e.*, if* $T \subseteq S$ *and* $\overline{T}$ *is an extension of* X*, then we have* $T = S$*.*

The set S depicted in Fig. 1d is a cut for X (Fig. 1a). It can be verified that $\overline{S}$ (Fig. 1c) is an extension of X and that S is minimal for this property.
If X is a subgraph of G and S a cut for X, it may be easily seen that $\overline{S}$ is a maximal extension of X.

The notion of graph cut has been studied for many years and is often defined by means of partitions. In this case, a set $S \subseteq E$ is said to be a graph cut if there exists a partition of V such that S is the set of all edges of G whose extremities are in two distinct sets of the partition. If each set of the partition is connected and contains the vertex set of a unique component of a subgraph of G, then S is a cut for this subgraph. It may be easily seen that this definition is equivalent to Def. 2. One of the most fundamental results in combinatorial optimization involves graph cuts. It states that given two isolated vertices of an edge-weighted graph (called source and sink), finding a cut of minimal cost that separates these vertices is equivalent to finding a maximum flow (see, for instance, [16], chapter 6.2). There exist polynomial-time algorithms to find the so-called min-cut. On the other hand, finding a cut of minimal cost among all the cuts for a subgraph which is not reduced to two isolated vertices is NP-hard [18]. In the forthcoming sections, we introduce the watershed cuts of an edge-weighted graph and show that these watersheds are graph cuts which also satisfy an optimality property. A major advantage is that they can be computed in linear-time.

2.2 Watersheds

We introduce the watershed cuts of an edge-weighted graph. To this aim, 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.

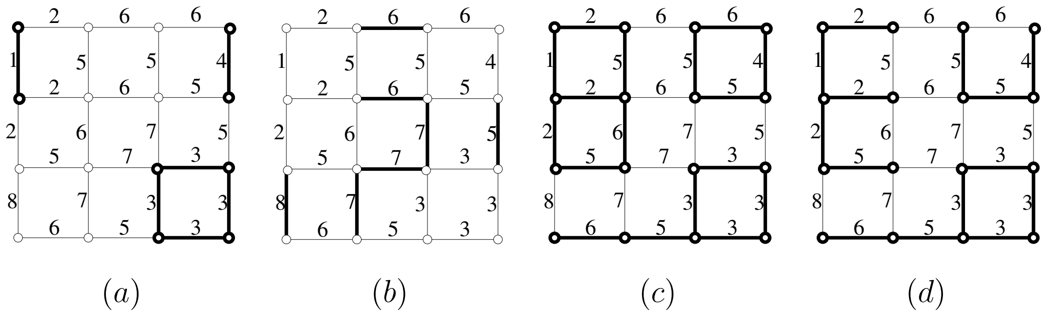


Fig. 2. A graph G and a map F. Edges and vertices in bold depict: (a), the minima of F; (b), a watershed S of F; (c), an extension of $M(F)$ which is equal to $\overline{S}$; and (d), a MSF relative to $M(F)$.

Let $\pi = \langle x_0, \ldots, x_l \rangle$ be a path in G. The path π is *descending (for F)* if, for any $i \in [1, l-1]$, $F(\{x_{i-1}, x_i\}) \geq F(\{x_i, x_{i+1}\})$.

Definition 3 (drop of water principle) *Let $S \subseteq E$. We say that S* satisfies the drop of water principle (for F) *if $\overline{S}$ is an extension of $M(F)$ and if for any $u = \{x_0, y_0\} \in S$, there exist $\pi_1 = \langle x_0, \ldots, x_n \rangle$ and $\pi_2 = \langle y_0, \ldots, y_m \rangle$ which are two descending paths in $\overline{S}$ such that:*
- x_n and y_m are vertices of two distinct minima of F; and
- $F(u) \geq F(\{x_0, x_1\})$ (resp. $F(u) \geq F(\{y_0, y_1\})$), whenever π_1 (resp. π_2) is not trivial.
If S satisfies the drop of water principle, we say that S is a watershed cut, or simply a watershed, *of F.*

We illustrate the previous definition on the function F depicted in Fig. 2. The function F contains three minima (Fig. 2a). We denote by S the set of bold edges depicted in Fig. 2b. It may be seen that $\overline{S}$ (Fig. 2c) is an extension of $M(F)$. Let $u = \{x, y\} \in S$ be the edge at altitude 8. There exists π_1 (resp. π_2) a descending path in $\overline{S}$ from x (resp. y) to the minimum at altitude 1 (resp. 3). The first edge of π_1 (resp. π_2) is lower than u whose altitude is 8. It can be verified that the previous properties hold for any edge in S. Thus, S is a watershed of F. The next statement follows from the definition of a watershed.

Property 4 *Let $S \subseteq E$. If S is a watershed of F, then S is a cut for $M(F)$.*

Notice that a watershed of F is defined thanks to conditions that depend of the altitude of the edges whereas the definition of a cut is solely based on the structure of the graph. Consequently, the converse of Prop. 4 is, in general, not true.

As an illustration of the previous property, it may be verified that the watershed of the map F, depicted in Fig. 2b, is a cut for the minima of F.

A popular alternative to Def. 3 defines a watershed exclusively by its catchment basins and does not involve any property of the divide [19–22]. In a vertex-weighted graph, such definitions raise several problems. The catchment basin of a minimum M can be defined as the points from which M can be reached by a path with steepest descent. In this case, several catchment basins may overlap each other. To avoid this problem, some authors define the catchment basin of M as the set of points from which M is the only minimum that can be reached by a path with steepest descent. In this case, some thick sets of points may not belong to any catchment basin (such situations are illustrated in [22]).

In the framework of edge-weighted graph, we define a *catchment basin* as a component of the complementary of a watershed. The following theorem (Th. 6) shows that a watershed can be defined equivalently by its divide line or by its catchment basins.

For that purpose, we start with some definitions relative to the notion of path with steepest descent.

From now on, we will also denote by F the map from V to $\mathbb{Z}$ such that for any $x \in V$, $F(x)$ is the minimal altitude of an edge which contains x, i.e.*, $F(x) = \min\{F(u) \mid u \in E, x \in u\}$; $F(x)$ is the* altitude *of x.*

Let $\pi = \langle x_0, \ldots, x_l \rangle$ be a path in G. The path π is *a path with steepest descent for F* if, for any $i \in [1, l]$, $F(\{x_{i-1}, x_i\}) = F(x_{i-1})$.

Definition 5 (steepest descent) *Let $S \subseteq E$ be a cut for $M(F)$. We say that S is a* basin cut *of F if, from each point of V to $M(F)$, there exists, in the graph induced by $\overline{S}$, a path with steepest descent for F.*

Theorem 6 (consistency) *Let $S \subseteq E$. The set S is a basin cut of F if and only if S is a watershed cut of F.*

The previous theorem establishes the consistency of watershed cuts: 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, in the literature about discrete watershed, no similar property has ever been proved. Some counter examples which show that such a duality does not hold in other frameworks can be found in [23]. Th. 6 thus emphasizes that the framework considered in this paper is adapted for the definition and study of discrete watersheds.

3 Minimum spanning forests and watershed optimality

In this section, we establish the optimality of watersheds. To this aim, we introduce the notion of minimum spanning forests relative to subgraphs of G. We will see that each of these forests induces a unique graph cut. The main result of this section (Th. 10) states that a graph cut is induced by a minimum spanning forest relative to the minima of a map if and only if it is a watershed of this map. In Sec. 3.2, we show that the problem of finding a relative minimum spanning forest is equivalent to the classical problem of finding a minimum spanning tree. In fact, this provides a mean to derive, from any minimum spanning tree algorithm, an algorithm for relative minimum spanning forests, and thus also, for watersheds.

Let X and Y be two non-empty subgraphs of G. We say that Y is a *forest relative to* X if:
i) Y is an extension of X; and
ii) for any extension $Z \subseteq Y$ of X, we have $Z = Y$ whenever $V(Z) = V(Y)$.
We say that Y is a *spanning forest relative to* X *(for* G*)* if Y is a forest relative to X and $V(Y) = V$.

Let $X \subseteq G$. We say that X is a *tree* (resp. a *spanning tree*) if X is a forest (resp. spanning forest) relative to the subgraph $(\{x\}, \emptyset)$, x being any vertex of X. We say that X is a *forest* (resp. a *spanning forest*) if X is a forest (resp. a spanning forest) relative to $(S, \emptyset)$, S being a subset of $V(X)$.

Remark 7 *The notions of tree and forest (resp. spanning tree and forest) presented above corresponds exactly to the usual notions of tree and forest (resp. spanning tree and forest). On the one hand, the notion of forest (resp. tree) is usually defined as a graph (resp. connected graph) which does not contain any cycle,* i.e., *any simple path whose first and last points are adjacent. On the other hand, it may be seen that a graph* X *is a forest relative to a subgraph* Y *of* G *if and only if* X *is an extension of* Y *and any cycle in* X *is also a cycle in* Y*. Thus, the two notions of forest (hence tree) are equivalent.*

Let X be a subgraph of G, *the weight of* X *(for* F*)* is the value $F(X) = \sum_{u \in E(X)} F(u)$.

Definition 8 *Let* X *and* Y *be two subgraphs of* G*. We say that* Y is a minimum spanning forest (MSF) relative to X (for F, in G) *if* Y *is a spanning forest relative to* X *and if the weight of* Y *is less than or equal to the weight of any other spanning forest relative to* X*. In this case, we also say that* Y *is a* relative MSF*.*

Let us consider the graph G depicted in Fig. 3 and the subgraph X depicted in bold in Fig. 3a. The graphs Y and Z (bold edges and vertices) in Figs. 3b and c are two MSFs relative to X.

3.1 Relative MSFs and watersheds

We now have the mathematical tools to present the main result of this section (Th. 10) which establishes the optimality of watersheds. It shows the equivalence between the cuts which satisfy the drop of water principle and those induced by the MSFs relative to the minima of a map.

We start by the following lemma which gives, thanks to Th. 6, a first intuition of Th. 10.

Lemma 9 *Let X be a spanning forest relative to $M(F)$. The graph X is a MSF relative to $M(F)$ if and only if, for any x in V, there exists a path in X from x to $M(F)$ which is a path with steepest descent for F.*

Let X be a subgraph of G and let Y be a spanning forest relative to X. There exists a unique cut S for Y and this cut is also a cut for X. We say that this unique cut is the *cut induced by* Y. Furthermore, if Y is a MSF relative to X, we say that that S is a *MSF cut for* X.

Theorem 10 (optimality) *Let $S \subseteq E$. The set S is MSF cut for $M(F)$ if and only if S is a watershed cut of F.*

As far as we know, this is the first result which establishes watershed optimality.

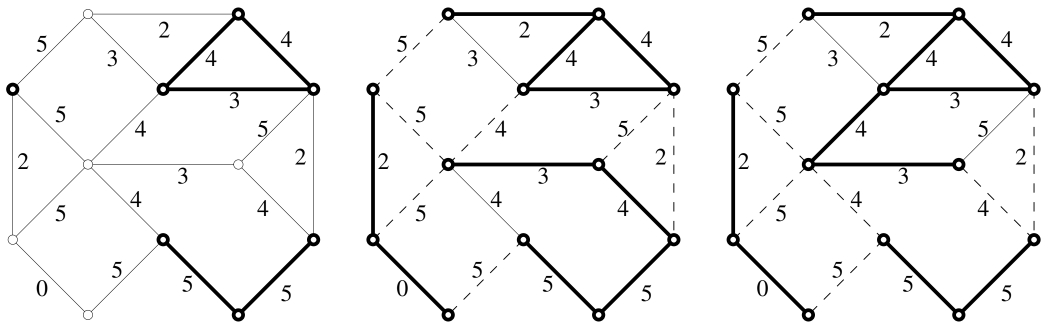
3.2 Relative MSFs and minimum spanning trees

The minimum spanning tree problem is one of the most typical and well-known problems of combinatorial optimization (see [24–27]). It has been applied for many years in image analysis [28]. We show that the minimum spanning tree problem is equivalent to the problem of finding a MSF relative to a subgraph of G.

Let $X \subseteq G$. The graph *X is a minimum spanning tree (for F, in G)* if X is a MSF relative to the subgraph $(\{x\}, \emptyset)$, x being any vertex of X.

Consequently to Rem. 7, it may be easily seen that the notion of minimum spanning tree presented above corresponds exactly to the usual one.

In order to recover the link between flooding algorithms and minimum spanning trees, in [11], F. Meyer proposed a construction which allows to show



(a) (b) (c)

Fig. 3. A graph G and a map F. The bold edges and vertices represent: (a), X a subgraph of G; (b) and (c), two MSFs relative to X; their induced cuts are represented by dashed edges.

the equivalence between finding a MSF rooted in a set of vertices and finding a minimum spanning tree. Here, we extend this construction for proving the equivalence between finding a minimum spanning tree and a MSF relative to a subgraph of G. Let us consider, in a first time, a graph $X \subseteq G$ such that $E(X) = \emptyset$, *i.e.*, a graph composed of isolated vertices. From G and X, we can construct a new graph $G' = (V', E')$ which contains an additional vertex z (*i.e.*, $z \notin V$) linked by an edge to each vertex of X. In other words, $V' = V \cup \{z\}$ and $E' = E \cup E_z$, where$E_z = \{\{x, z\} \mid x \in V(X)\}$. Let us consider the map F' from E' to $\mathbb{Z}$ such that, for any $u \in E$, $F'(u) = F(u)$ and for any $u \in E_z$, $F'(u) = k_{\min} - 1$, $k_{\min}$ being the minimum value of F. Let Y be any subgraph of G and let Y' be the graph whose vertex and edge sets are respectively $V(Y) \cup \{z\}$ and $E(Y) \cup E_z$. It may be seen that Y' is a minimum spanning tree for F' in G' if and only if Y is a MSF relative to X for F in G.
The construction presented above can be easily generalized to any subgraph X of G. To this aim, in a preliminary step, each component of X must be contracted into a single vertex and, if two vertices of the contracted graphs must be linked by multiple edges, only the one with minimal value is kept.

A direct consequence of the construction presented above is that any minimum spanning tree algorithm can be used to compute a relative MSF. Many efficient algorithms (see a survey in [29]) exist in the literature for solving the minimum spanning tree problem. In particular, in a recent paper [30], B. Chazelle proposed a quasi-linear time algorithm. In the sequel of this paper, we will see that a better complexity can be reached to compute MSFs relative to the minima of F.

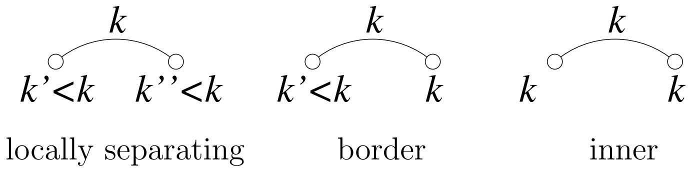


Fig. 4. Illustration of the different local configurations for edges.

4 Optimal thinnings

As seen in the previous section, a MSF relative to a subgraph of G can be computed by any minimum spanning tree algorithm. The best complexity for solving this problem is reached by the quasi-linear algorithm of Chazelle [30]. In this section, we introduce a new paradigm to compute MSFs relative to the minima of a map and obtain a linear algorithm. To this aim, we define a new thinning transformation that iteratively lowers the values of the edges that satisfy a simple local property. The minima of the transformed map constitute precisely a MSF relative to the minima of the original one. More remarkably, any MSF relative to the minima of a map can be obtained by this transformation. We discuss the possibility of parallel algorithms based on this transformation and give a sequential implementation (Algo. 1) which runs in linear time.

4.1 Border thinnings and watersheds

We introduce an edge classification based exclusively on local properties, *i.e.*, properties which depend only on the adjacent edges. This classification will be used in the definition of a lowering process (Def. 12) which allows to extract the watersheds of a map.

Remind that, if x is a vertex of G, $F(x)$ is the minimal altitude of an edge which contains x.

Definition 11 *Let $u = \{x, y\} \in E$.*
We say that u is locally separating (for F) *if $F(u) > \max(F(x), F(y))$.*
We say that u is border (for F) *if $F(u) = \max(F(x), F(y))$ and $F(u) > \min(F(x), F(y))$.*
We say that u is inner (for F) *if $F(x) = F(y) = F(u)$.*

Fig. 4 illustrates the above definitions. In Fig. 5a, $\{j, n\}$, $\{a, e\}$ and $\{b, c\}$ are examples of border edges; $\{i, m\}$ and $\{k, l\}$ are inner edges and both $\{h, l\}$ and $\{g, k\}$ are locally-separating edges. Note that any edge of G corresponds exactly to one of the types presented in Def. 11.

Let $u \in E$. The *lowering of F at u* is the map F' in $\mathcal{F}$ such that:

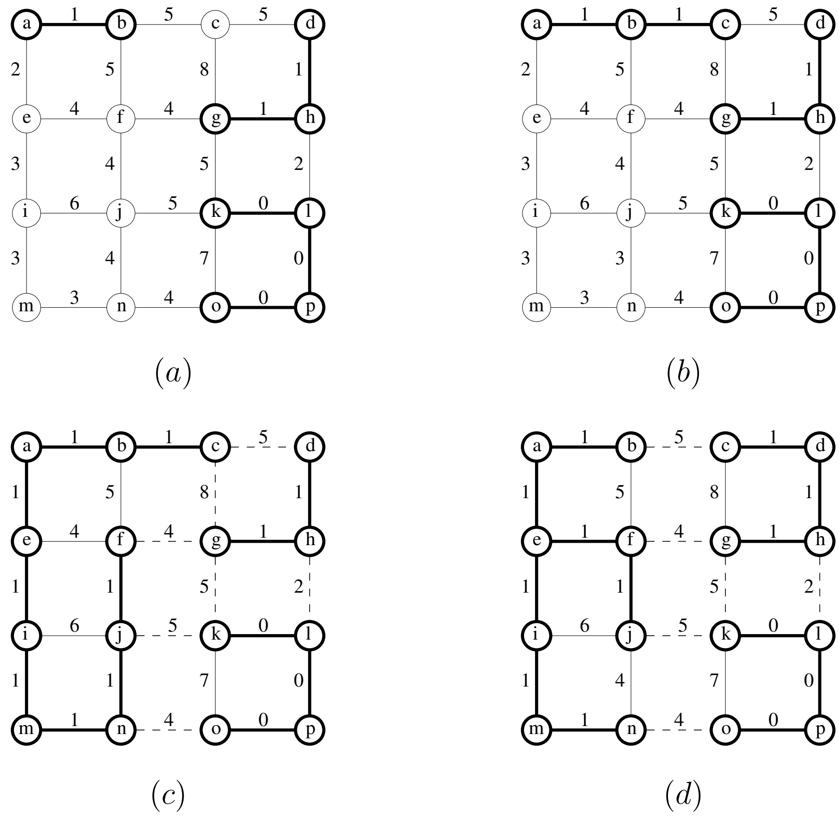


Fig. 5. A graph and some associated functions. The bold graphs superimposed are the minima of the corresponding functions; (b), a border thinning of (a); (c): a border kernel of both (a) and (b); and (d), another border kernel of (a). In (c) and (d), the border cuts are represented by dashed edges.

- $F'(u) = \min_{x \in u}\{F(x)\}$; and
- $F'(v) = F(v)$ for any edge $v \in E \setminus \{u\}$.

Definition 12 (border cut) *Let $H \in \mathcal{F}$. We say that H is a* border thinning of F *if:*
i) $H = F$; or
ii) there exists $I \in \mathcal{F}$ a border thinning of F such that H is the lowering of I at a border edge for I.
If there is no border edge for H, we say that H is a border kernel*. If H is a border thinning of F and if it is a border kernel, we say that H is a* border kernel of F.
If H is a border kernel of F, any cut for $M(H)$ is called a border cut for F.

To illustrate the previous definition, we assume that F (resp. H, I) is the map of Fig. 5a (resp. b,c). The maps H and I are border thinnings of F. The map I is a border kernel of both F and H. The function depicted in Fig. 5d is another border kernel of F which is not a border kernel of H. In Fig. 5c and d, the border cuts are represented by dashed edges. Remark that the minima of the two border kernels constitute forests relative to $M(F)$, and that all edges which do not belong to the bold graphs are locally separating.

We now present an important result of this section which mainly states that the border kernels can by used to compute MSFs relative to the minima of a map.

Property 13 *Let $H \in \mathcal{F}$. If H is a border thinning of F, then any MSF relative to $M(H)$ (for H) is a MSF relative to $M(F)$ (for F). Furthermore, if H is a border kernel of F, then $M(H)$ is itself a MSF relative to $M(F)$ (for F).*

In other words, the border thinning transformation preserves some MSF relative to the minima of the original map and, moreover, the border kernels allow the extraction of MSFs relative to the minima. We remind that a MSF relative to the minima of a map defines a unique cut which is a watershed of this map. Thus, a border kernel of a map defines a unique border cut for this map. Hence, from Prop. 13, we can easily prove the following corollary which links border kernels to watershed cuts.

Corollary 14 *Any border cut of F is a watershed cut of F.*

Thanks to classical algorithms for minima computation, a MSF relative to $M(F)$ can be obtained from any border kernel of F. In fact, the minima of a border kernel can be extracted by a much more efficient strategy.

Property 15 *Let F be a border kernel. An edge u is in a minimum of F if and only if u is inner for F.*

Let H denote a border kernel of F. On the one hand, the map H and its minima can be derived from F exclusively by *local operations* (see Defs. 11, 12 and Prop. 15). On the other hand, a MSF relative to $M(F)$ is a *globally optimal structure*. The minima of H constitute, by Prop. 13, a MSF relative to $M(F)$. Thus, the local and order-independent operations presented in this section always produce a globally optimal structure.

This kind of local, order-independent operations can be efficiently exploited by dedicated hardware. For instance, raster scanning strategies for extracting a border cut can be straightforwardly derived. It has been shown that such strategies can be very fast on adapted hardware [31].

As mentioned above the property of a border edge can be tested locally. Thus, if a set of *independent* (*i.e.*, mutually disjoint) border edges is lowered in parallel, then the resulting map is a border thinning. This property offers several possibilities of parallel watershed algorithms. In particular, efficient algorithms for array processors can be derived.

On a sequential computer, a naive algorithm to obtain a border kernel could be the following: *i*) for all $u = \{x, y\}$ of G, taken in an arbitrary order, check the values of $F(u)$, $F(x)$ and $F(y)$ and whenever u is border, lower the value of u down to the minimum of $F(x)$ and $F(y)$ (cost for checking all edges of G: $O(|E|)$); *ii*) repeat step *i*) until no border edge remains. Consider the graph G whose vertex set is $\{0, \dots, n\}$ and whose edge set is made of all the pairs $u_i = \{i, i+1\}$ such that $i \in [0, n-1]$. Let $F(u_i) = n-i$, for all $i \in [0, n-1]$. On this graph, if the edges are processed in the order of their indices, step *i*) will be repeated exactly $|E|$ times. The worst case time complexity of this naive algorithm is thus at least $O(|E|^2)$. In order to reduce the complexity, we introduce a second lowering process in which any edge is lowered at most once. This process is a particular case of border thinning which also produces, when iterated until stability, a border kernel of the original map. Thanks to this second thinning strategy, we derive a linear-time algorithm to compute border kernels and, thus, watersheds.

It may be seen that an edge which is in a minimum at a given step of a border thinning sequence never becomes a border edge. Thus, lowering first the edges adjacent to the minima seems to be a promising strategy. In order to study and understand this strategy, we may classify any inner, border or locally-separating edge with respect to the adjacent minima. We thus obtain the 8 cases illustrated in Fig. 6 and any edge is classified in exactly one of these classes depending on the values of its adjacent edges and on the regional minima. In this section we study, in particular, a transformation which iteratively lowers the values of the border edges adjacent to minima (see Fig. 6F).

Definition 16 *We say that an edge u in E is* minimum-border (for F), written M-border*, if u is border for F and if exactly one of the vertices in u is a vertex of $M(F)$.*

In Fig. 5a, the edge $\{c, d\}$ is M-border whereas $\{j, n\}$ is not.

Definition 17 (M-border cut) *Let $H \in \mathcal{F}$. We say that H* is an M-border thinning of F *if:*
i) *$H = F$; or*
ii) *there exists $I \in \mathcal{F}$ an M-border thinning of F such that H is the lowering of I at an M-border edge for I.*
If there is no M-border edge for H, we say that H is an M-border kernel*. If H is an M-border thinning of F and if it is an M-border kernel, we say that H is an* M-border kernel of F*.*
If H is an M-border kernel of F, any cut for $M(H)$ is called an M-border cut for F.

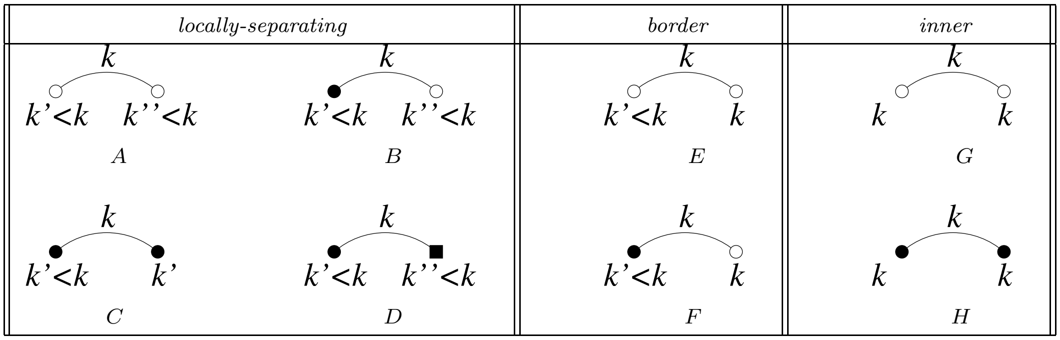


Fig. 6. Edge-classification in a weighted graph. In the figure, any black vertex belongs to a minimum and two vertices represented by different shapes (*i.e.*, square and circle) belong to distinct minima.

Observe (in Fig. 5ab for instance) that when a map is lowered at an M-border edge, one of its minima is “growing by one vertex”. The well-known watershed algorithms by simulation of a flooding [4, 6, 11] (see also [32]) also work as “region growing” processes. Intuitively, for a vertex-weighted graph, they can be described as follows: (*i*), mark the minima with distinct labels; (*ii*), mark the lowest point adjacent to a single label with this label; and (*iii*), repeat step (*ii*) until there is no point adjacent to a single label. In fact, as stated by the following property, the M-border thinning transformation –which is itself a particular case of border thinning– generalizes flooding algorithms.

If u is an edge with minimal altitude among all the edges outgoing from $M(F)$, then u is a *flooding edge for* F.

Property 18 *Any flooding edge for* F *is an M-border edge for* F*.*

The previous property also establishes a link with one [26] of the famous minimum spanning tree algorithms. Let us consider the construction presented in Sec. 3.2 for the computation of a MSF relative to $M(F)$. In this case, as stated by F. Meyer in [11], the edges considered by Prim’s algorithm are exactly those considered in a lowering sequence using the above condition to detect M-border edges. Prop. 18 thus gives us a clue to precisely determine the relation between M-border thinnings and MSFs relative to the minima. To this aim, let us formalize the notions of flooding kernel and cut.

Definition 19 (flooding cut) *Let* $H \in \mathcal{F}$*. We say that* H *is a* flooding of F *if* $H = F$ *or if there exists a flooding* I *of* F *such that* H *is the lowering of* I *at a flooding edge for* I*.*
If there is no flooding edge for H*, we say that* H *is a* flooding kernel*. If* H *is a flooding of* F *and if it is a flooding kernel, we say that* H *is a* flooding kernel of F*. In this case, any cut for* $M(H)$ *is called a* flooding cut for F*.*

Let $H \in \mathcal{F}$. We say that $M(H)$ is the *min-graph of* H. This notion will be used in the following property which states that, as well as the min-graphs of border kernels, the min-graphs of M-border and flooding kernels of F are MSFs relative to $M(F)$. More remarkably, any MSF relative to $M(F)$ can be obtained as the min-graph of an M-border kernel, of a flooding kernel and also as the min-graph of a border kernel of F.

Lemma 20 *Let $X \subseteq G$. The four following statements are equivalent:*
(i) X is the min-graph of a flooding kernel of F;
(ii) X is the min-graph of an M-border kernel of F;
(iii) X is the min-graph of a border kernel of F; and
(iv) X is a MSF relative to $M(F)$.

Since a relative MSF induces a unique graph cut, from the previous lemma, we immediately deduce that a flooding kernel (resp. an M-border kernel, a border kernel) of a map defines a unique flooding cut (resp. M-border cut, border cut). Hence, the following theorem which states the equivalence between watershed cuts, border cuts, M-border cuts and flooding cuts can be easily proved.

Theorem 21 *Let $S \subseteq E$. The four following statements are equivalent:*
(i) S is a flooding cut for F;
(ii) S is an M-border cut for F;
(iii) S is a border cut for F; and
(iv) S is a watershed cut for F.

Using the notions introduced in this section, we derive Algo. 1, an efficient algorithm to compute M-border kernels, hence watershed cuts. We recall that an edge u is border for F if the altitude of one of its extremities equals the altitude of u and the altitude of the other is strictly less than the altitude of u.

Algorithm 1: M-Border

Data: (V, E, F): an edge-weighted graph;
Result: F, an M-border kernel of the input map and M its minima.

$L \leftarrow \emptyset$;
Compute $M(F) = (V_M, E_M)$ and $F(x)$ for each $x \in V$;
foreach $u \in E$ *outgoing from* (V_M, E_M) **do** $L \leftarrow L \cup \{u\}$;
while *there exists* $u \in L$ **do**
  $L \leftarrow L \setminus \{u\}$;
  if *u is border for F* **then**
    $x \leftarrow$ the vertex in u such that $F(x) < F(u)$;
    $y \leftarrow$ the vertex in u such that $F(y) = F(u)$;
    $F(u) \leftarrow F(x)$; $F(y) \leftarrow F(u)$;
    $V_M \leftarrow V_M \cup \{y\}$; $E_M \leftarrow E_M \cup \{u\}$;
    foreach $v = \{y', y\} \in E$ *such that* $y' \notin V_M$ **do** $L \leftarrow L \cup \{v\}$;

In order to achieve a linear complexity, the graph (V, E) can be stored as an

array of lists which maps to each point the list of all its adjacent vertices. An additional mapping can be used to access in constant time the two vertices which compose a given edge. Nevertheless, for applications to image processing, and when usual adjacency relations are used, these structures do not need to be explicit.
Furthermore, to achieve a linear complexity, the minima of F must be known at each iteration. To this aim, in a first step (line 2), the minima of F are computed and represented by two boolean arrays V_M and E_M, the size of which are respectively $|V|$ and $|E|$. This step can be performed in linear time thanks to classical algorithms. Then, in the main loop (line 4), after each lowering of F (line 9), V_M and E_M are updated (line 10). In order to access, in constant time, the edges which are M-border, the (non-already examined) edges outgoing from the minima are stored in a set L (lines 3 and 11). This set can be, for instance, implemented as a queue. Thus, we obtain the following property.

Property 22 *At the end of Algorithm 1, F is an M-border kernel of the input function F. Furthermore Algorithm 1 terminates in linear time with respect to $|E|$.*

We emphasize that Algo. 1 does not require any sorting step nor the use of any hierarchical queue. Thus, whatever the range of the considered map, it runs in linear time with respect to the size of the input graph. To our best knowledge, this is the first watershed algorithm with such a property.

5 Streams and linear-time watershed algorithm

A watershed of any map can be computed in linear time thanks to Algo. 1. This algorithm, in a first step (line 2), extracts the minima of the input map. In this section, we introduce a second linear-time algorithm. As Algo. 1, it does not require any sorting step, nor the use of any hierarchical queue. Furthermore, contrarily to Algo. 1, it does not need minima precomputation. As far as we know, this algorithm has no equivalent in the literature.

In the first part of the section, the mathematical tools which allow to prove the correctness of the proposed algorithm are introduced. In particular, we propose a new notion of stream which is crucial to this paradigm. Then, the algorithm is presented, and both its correctness and complexity are analyzed.

Definition 23 *Let $L \subseteq V$. We say that L is a* stream *if, for any two points x and y of L, there exists, in L, either a path from x to y or from y to x, with steepest descent for F.*
Let L be a stream and let $x \in L$. We say that x is a top (resp. bottom) of L *if the altitude of x is greater than (resp. less than) or equal to the altitude of*

any $y \in L$.

Remark that if L is a stream and x is a bottom (resp. a top) of L, then, from any $y \in L$ to x (resp. from x to any $y \in L$), there is a path in L, with steepest descent for F. Notice that, whatever the stream L, there exists a top (resp. bottom) of L. Nevertheless, this top (resp. bottom) is not necessarily unique.

In order to illustrate the previous definitions, let us assume that G and F are the graph and the function depicted in Fig. 5a. The sets $L = \{a, b, e, i\}$ and $\{j, m, n\}$ are two examples of streams. On the contrary, the set $L' = \{i, j, k\}$ is not a stream since there is no path in L', between i and k, with steepest descent for F. The sets $\{a, b\}$ and $\{i\}$ are respectively the set of bottoms and tops of L.

The algorithm which will be proposed in this section is based on the iterative extraction of streams. In order to build such a procedure, we study stream concatenation.

Let L_1 and L_2 be two disjoint streams (*i.e.,* $L_1 \cap L_2 = \emptyset$) and let $L = L_1 \cup L_2$. We say that L_1 *is under* L_2, written $L_1 \prec L_2$, if there exist a top x of L_1, a bottom y of L_2, and there is, from y to x, a path in L with steepest descent for F. Note that, if $L_1 \prec L_2$, then L is also a stream.
We say that a stream L is an $\prec$-*stream* if there is no stream under L.

In Fig. 5a the stream $\{a, b, e, i\}$ is under the stream $\{j, m, n\}$ and thus $\{a, b, e, i, j, m, n\}$ is also a stream. Furthermore, there is no stream under $\{a, b, e, i\}$ and $\{a, b, e, i, j, m, n\}$. Thus, these are two $\prec$-streams.

The streams extracted by our algorithm are $\prec$-streams. As said in the introduction, this algorithm does not require minima precomputation. In fact, there is a deep link between $\prec$-streams and minima.

Property 24 *Let L be a stream. The three following statements are equivalent:*
(1) *L is an $\prec$-stream;*
(2) *L contains the vertex set of a minimum of F; and*
(3) *for any $y \in V \setminus L$ adjacent to a bottom x of L, $F(\{x, y\}) > F(x)$.*

In Fig. 5a the two $\prec$-streams $\{a, b, e, i\}$ and $\{a, b, e, i, j, m, n\}$ contain the set $\{a, b\}$ which is the vertex set of a minimum of F.
Remark that any stream L which contains an $\prec$-stream is itself an $\prec$-stream. We also notice that if L is an $\prec$-stream, the set of all bottoms of L constitutes the vertex set of a minimum of F. Furthermore, a subset L of V is the vertex set of a minimum of F if and only if it is an $\prec$-stream minimal for the inclusion relationship, *i.e.*, no proper subset of L is an $\prec$-stream.

In order to partition the vertex set of G, from the $\prec$-streams of F, the vertices of the graph can be arranged in the following manner.

Let $\mathcal{L} = \{L_1, \ldots, L_n\}$ be a set of n $\prec$-streams. We say that $\mathcal{L}$ is a *flow family* if:
- $\cup\{L_i \mid i \in \{1, \ldots, n\}\} = V$; and
- for any two distinct L_1 and L_2 in $\mathcal{L}$, if $L_1 \cap L_2 \neq \emptyset$, then $L_1 \cap L_2$ is the vertex set of a minimum of F.

Let $\mathcal{L}$ be a flow family and let $x \in V$. It may be seen that, either x belongs to a minimum of F (in this case, it may belong to several elements of $\mathcal{L}$), or x belongs to a unique $\prec$-stream of $\mathcal{L}$ which itself contains the vertex set of a unique minimum of F. Thus, thanks to $\mathcal{L}$, we can associate to each vertex x of G a unique minimum of F.

Definition 25 *Let $\mathcal{L}$ be a flow family. Let us denote by $M_1, \ldots, M_n$ the minima of F. Let ψ be the map from V to $\{1, \ldots, n\}$ which associates to each vertex x of V, the index (or label) i such that M_i is the unique minimum of F included in an $\prec$-stream of $\mathcal{L}$ which contains x; we say that ψ* a flow mapping *of F.*
If ψ is a flow mapping of F, we say that the set $S = \{\{x, y\} \in E \mid \psi(x) \neq \psi(y)\}$ is a flow cut *for F.*

The next proposed algorithm produces a flow mapping, hence a flow cut. The following theorem, which is a consequence of the definitions of flow families and basin cuts and of the consistency theorem, states the equivalence between flow cuts and watersheds. It constitutes the main tool to establish the correctness of Algo. 2.

Theorem 26 *Let $S \subseteq E$. The set S is a watershed of F if and only if S is a flow cut for F.*

We now present Algo. 2 which computes a flow mapping, hence, by Th. 26, a watershed. It iteratively assigns a label to each point of the graph. To this aim, from each non-labeled point x, a stream L composed of non-labeled points and whose top is x is computed (line 4). If L is an $\prec$-stream (line 5), a new label is assigned to the points of L. Otherwise (line 8), there exists an $\prec$-stream L_1 under L and which is already labeled. In this case, the points of L receive the label of L_1 (line 9). The function `Stream`, called at line 4, allows to compute the stream L. Roughly speaking, it performs an intermixed sequence of depth-first and breadth-first exploration of the paths with steepest descent. The main invariants of the function `Stream` are: i), the set L is, at each iteration, a stream; and ii), the set L' is made of all non-already explored bottoms of L. The function halts at line 17 when all bottoms of L have been explored or, at line 9, if a point z already labeled is found. In the former case, by Prop. 24, the returned set L is an $\prec$-stream. In the latter case, the label *lab* of z is also returned and there exists a bottom y of L such that $\langle y, z \rangle$ is a path with steepest descent. Thus, there is an $\prec$-stream L_1, under L, included in the set of all vertices labeled lab. Thus, by the preceding remarks, the output of Algo. 2 is a flow mapping of F. Furthermore, using the data structures presented in Sec. 4.2, we obtain a linear complexity.

Algorithm 2: Watershed

Data: (V, E, F): an edge-weighted graph;
Result: ψ: a flow mapping of F.

foreach $x \in V$ **do** $\psi(x) \leftarrow NO\_LABEL$;
$nb\_labs \leftarrow 0$; /* the number of minima already found */
foreach $x \in V$ *such that* $\psi(x) = NO\_LABEL$ **do**
  $[L, lab] \leftarrow$ `Stream`(V, E, F, ψ, x) ;
  if $lab = -1$ **then** /* L is an $\prec$-stream */
    nb_labs++ ;
    foreach $y \in L$ **do** $\psi(y) \leftarrow nb\_labs$;
  else
    foreach $y \in L$ **do** $\psi(y) \leftarrow lab$;

Function `Stream(` V, E, F, ψ, x`)`

Data: (V, E, F): an edge-weighted graph; ψ: a labeling of V; x: a point in V.
Result: $[L, lab]$ where L is a stream such that x is a top of L, and lab is either the label of an $\prec$-stream under L, or -1.

$L \leftarrow \{x\}$;
$L' \leftarrow \{x\}$; /* the set of non-explored bottoms of L */
while *there exists* $y \in L'$ **do**
  $L' \leftarrow L' \setminus \{y\}$;
  $breadth\_first \leftarrow TRUE$;
  while *(breadth_first) and (there exists* $\{y, z\} \in E$ *such that* $z \notin L$ *and* $F(\{y, z\}) = F(y)$*)* **do**
    if $\psi(z) \neq NO\_LABEL$ **then**
      /* there is an $\prec$-stream under L already labelled */
      return $[L, \psi(z)]$;
    else if $F(z) < F(y)$ **then**
      $L \leftarrow L \cup \{z\}$; /* z is now the only bottom of L */
      $L' \leftarrow \{z\}$; /* hence, switch to depth-first exploration */
      $breadth\_first \leftarrow FALSE$;
    else
      $L \leftarrow L \cup \{z\}$; /* $F(z) = F(y)$, thus z is also a bottom of L */
      $L' \leftarrow L' \cup \{z\}$; /* continue breadth-first exploration */
return $[L, -1]$;

Property 27 *Algorithm 2 outputs a map ψ which is a flow mapping of F. Furthermore, Algorithm 2 runs in linear-time with respect to $|E|$.*

Remark that, in function `Stream`, the use of breadth-first iterations is required to ensure that the produced set L is always an $\prec$-stream. Otherwise, if only depth-first iterations were used, `Stream` could be stuck on plateaus (*i.e.*, connected subgraphs of G with constant altitude) since some bottoms of L would never be explored.

6 Watersheds, shortest-path forests and topological watersheds

An interesting feature of the framework settled in this paper is that it allows to understand the links and differences between several methods used in image segmentation. Thanks to relative MSFs and border kernels, we provide a mathematical comparison between watershed cuts and:

- shortest-path forests (the theoretical basis of the Image Foresting Transform (IFT) [12] and the fuzzy connected image segmentation [15, 33]); and
- topological watersheds [7, 8].

6.1 Connection Value

The connection value [8, 10, 34] (also called degree of connectivity [35] or fuzzy connectedness [14] up to an inversion of F) may be seen as a measure of contrast between two subgraphs. This notion is central to both topological watersheds and shortest-path forests. We begin the section by defining the connection value. Then, we show that any MSF relative to a subgraph of G "preserves" the connection values.

Definition 28 *Let $\pi = \langle x_0, \ldots, x_l \rangle$ be a path in G. If π is non-trivial, we set $F(\pi) = \max\{F(\{x_{i-1}, x_i\}) \mid i \in [1, l]\}$. If π is trivial, we set $F(\pi) = F(x_0)$. Let X and Y be two subgraphs of G, we denote by $\Pi(X, Y)$ the set of all paths from X to Y in G. The* connection value between X and Y (in G and for F) *is $F(X, Y) = \min\{F(\pi) \mid \pi \in \Pi(X, Y)\}$.*

Let X be any subgraph of G. The following theorem asserts that, if the connection value between two components of X is equal to k, then the connection value between the two corresponding components in any MSF relative to X is also k: relative MSFs preserve the connection values.

Theorem 29 *Let X be a subgraph of G. If Y is a MSF relative to X, then for any two distinct components A and B of X, we have $F(A, B) = F(A', B')$,* *where A' and B' are the two components of Y such that $A \subseteq A'$ and $B \subseteq B'$.*

For example, in Fig. 5a, the connection value between the two minima at altitude 1 is equal to 4. It can be verified that the connection value between the two corresponding components of the MSFs relative to the minima, depicted in Figs. 5c and d, is also 4.

Let $S \subseteq E$ be a watershed cut of F. As a corollary of Th. 29, it may be deduced that the connection value between two distinct catchment basins (*i.e.*, two components of $\overline{S}$) is equal to the connection value between the two corresponding minima of F. To put it briefly, watershed cuts preserve the connection value.

6.2 Shortest-path forests

We investigate the links between relative MSFs and shortest-path forests. The image foresting transform [12], the inter-pixel flooding watershed [4, 6], and the relative fuzzy connected image segmentation [14,15,33,36] fall in the scope of shortest-path forests. Roughly speaking these methods partition the graph into connected components associated to seed points. The component of each seed consists of the points that are "more closely connected" to this seed than to any other. In many cases, in order to define the relation is "more closely connected to", the chosen measure is precisely the connection value, *i.e.*, a path π' is considered as shorter than a path π whenever $F(\pi') < F(\pi)$. The resulting regions are then given by a shortest-path forest. We show that any MSF relative to a subgraph X is a shortest-path spanning forest relative to X and that the converse is not true. Furthermore, we prove that both concepts are equivalent whenever X corresponds to the minima of the considered map.

If x is a vertex of G, to simplify the notation, the graph $(\{x\}, \emptyset)$ will be also denoted by x. Let X and Y be two subgraphs of G. We say that Y is a *shortest-path forest relative to* X if Y is a forest relative to X and if, for any $x \in V(Y)$, there exists, from x to X, a path π in Y such that $F(\pi) = F(x, X)$. If Y is a shortest-path forest relative to X and $V(Y) = V$, we say that Y is a *shortest-path spanning forest relative to* X. In this case, the unique cut induced by Y is a *SPF cut for* X.

Let G be the graph in Fig. 7 and let F be the corresponding map. Let X, Y, Z be the bold graphs depicted in respectively Figs. 7a,b and c. The two graphs Y and Z are shortest-path spanning forests relative to X.

Property 30 *Let X and Y be two subgraphs of G. If Y is a MSF relative to X, then Y is a shortest-path spanning forest relative to X. Furthermore, any MSF cut for X is a SPF cut for X.*

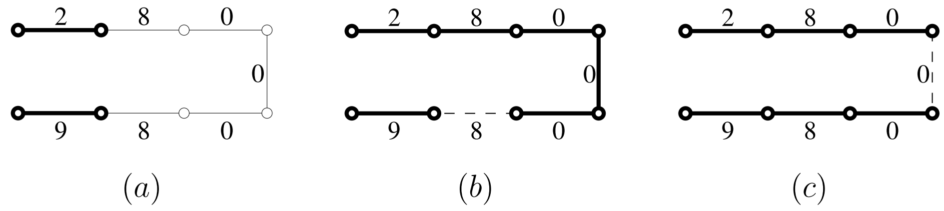


Fig. 7. A graph G and a function F. The bold edges and vertices represent: (a), a graph X; (b), a MSF relative to X; (c), a shortest-path spanning forest relative to X which is not a MSF relative to X.

The converse of Prop. 30 is, in general, not true. For example, the graph Z (Fig. 7c), is a shortest-path spanning forest relative to the graph X (Fig. 7a) whereas it is not a MSF relative to this graph. On the same example (Fig. 7c) we can also observe that, contrarily to relative MSFs, shortest-path spanning forests do not always preserve the connection value (in the sense of Th. 29). In particular, in Fig. 7, the connection value between the two components of X is equal to 8, whereas the connection value between the two components of Z is equal to 0. Then, on the contrary of cuts induced by relative MSFs (see for instance Fig. 7b), the cuts induced by shortest-path spanning forests are not necessarily located on the "crests" of the function.

In fact, as stated by the following property, if the graph X constitutes precisely the minima of F, the equivalence between both concepts can be established.

Property 31 *Let X be a subgraph of G. A necessary and sufficient condition for X to be a shortest-path spanning forest relative to $M(F)$ is that X is a MSF relative to $M(F)$. Furthermore, a subset of E is a MSF cut for $M(F)$ if and only if it is a SPF cut for $M(F)$.*

Whereas the notions of shortest-path forests and relative MSFs are equivalent when extensions of the minima are considered, in the general case, the relative MSFs satisfy additional interesting properties, such as the preservation of the connection value or the optimality (in the sense of Def. 8). Relative MSFs thus constitute a method of choice for marker (seed) based segmentation procedures, an illustration of which is provided in Sec. 7.2.

6.3 Topological watershed

The notions of a W-thinning and a topological watershed, introduced in [7], allow to rigorously define the watersheds of a vertex-weighted graph and to prove important properties (see [8–10]) not guaranteed by most popular watershed algorithms for vertex-weighted graphs. In particular, in [8, 10], the equivalence between a class of transformations which preserves the connection value and the W-thinnings is proved. Thus, Th. 29 invites us to recover the links between watershed cuts and topological watersheds.

The W-thinnings and topological watersheds are defined for graphs whose vertices are weighted by a cost function (denoted by I in the following). For that purpose, we introduce a minimal set of definitions to handle this framework.

Let $P \subseteq V$. The *subgraph of* G *induced by* P, denoted by G_P, is the graph whose vertex set is P and whose edge set is made of all edges of G linking two points in P, *i.e.*, $G_P = (P, \{\{x, y\} \in E \mid x \in P, y \in P\})$. Let I be a map from V to $\mathbb{Z}$, and let $k \in \mathbb{Z}$. We denote by $\overline{I}[k]$ the subgraph of G induced by the set of all points $x \in V$ such that $I(x) < k$; $\overline{I}[k]$ is called a *(level k) lower-section* of I.

Definition 32 *Let I be a map from V to $\mathbb{Z}$.*
Let x in V and $k = I(x)$. If x is adjacent to exactly one component of $\overline{I}[k]$, we say that x is W-destructible for I.
Let J be a map from V to $\mathbb{Z}$. We say that J is a W-thinning of I (in G) *if $J = I$ or if J may be derived from I by iteratively lowering the values of W-destructible points by one.*
We say that J is a topological watershed of I *if J is a W-thinning of I and if there is no W-destructible point for J.*

Let us consider the map I depicted in Fig. 8d. The points at altitude 2 are both W-destructible whereas the point at altitude 5 is not. The maps J and K depicted, respectively, in (e) and (f) are W-thinnings of F. The interested reader can verify that there exists a sequence of maps which allows to obtain J (resp. K) from I by iteratively lowering by one the values of W-destructible points. Notice that J is a topological watershed of I, since there is no W-destructible point for J. On the other hand, K is not a topological watershed of I, indeed the points with altitude 10, 6 or 4 are W-destructible.

We now present the notion of line graph (see [37], and [32,38,39] for topological watershed properties in this framework). This concept provides a simple way to automatically infer definitions and properties from vertex-weighted graphs to edge-weighted graphs.

Definition 33 *The* line graph of $G = (V, E)$ *is the graph* (E, Γ)*, such that* $\{u, v\}$ *belongs to* Γ *whenever* $u \in E$*,* $v \in E$*, and* u *and* v *are adjacent,* i.e., $|u \cap v| = 1$.

To each graph G whose edges are weighted by a cost function F, we can associate its line graph G'. The vertices of G' are weighted by F and thus any transformation of F can be performed either in G or in G'. Fig. 8 illustrates such a procedure. Let G be the graph depicted in (a), (b) and (c). The line graph of G is depicted in (d), (e) and (f). The map shown in (b) and (e) is a topological watershed of the one shown in (a) and (d). The map in (c) and (f)

is a border kernel.

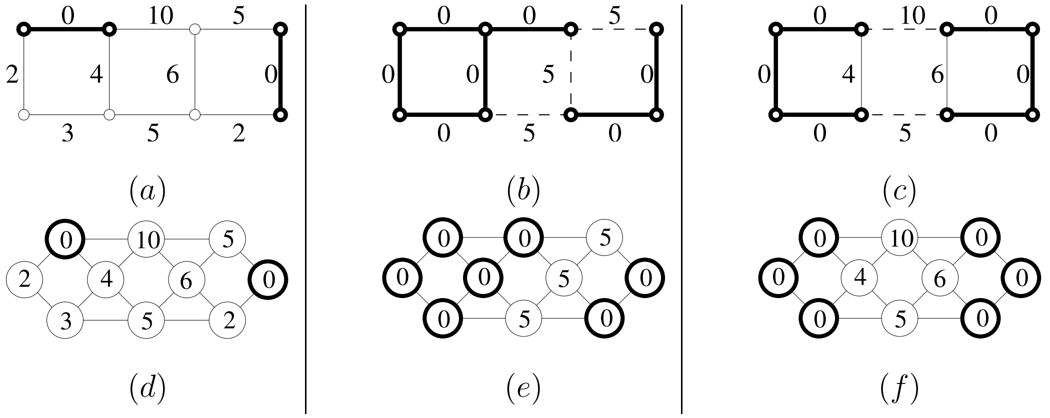


Fig. 8. Illustration of line graphs and topological watersheds. The graph in (d) (resp. (e), (f)) is the line graph of the one in (a) (resp. (b), (c)). The minima of the associated functions are depicted in bold. (c, f): the map is a border kernel of the one in (a, d); (b, e) and (c, f): the maps are W-thinnings of (a, d); the map in (b, e) is furthermore a topological watershed of the one in (a, d).

Definition 34 *Let $S \subseteq E$. We say that S is a* topological cut for F, *if there exists a W-thinning H of F in the line graph of G such that S is the only cut for $M(H)$.*

Property 35 *Let $H \in \mathcal{F}$. If H is a border thinning of F in G, then H is a W-thinning of F in the line graph of G. Furthermore, any border cut for F is a topological cut for F.*

The previous property is illustrated in Fig. 8 where the map depicted in Fig. 8c is a border thinning of F (Fig. 8a), thus a W-thinning of F. The converse of Prop. 35 is not true. The map H (Fig. 8b) is a topological watershed of F but it is not a border kernel of F. Indeed, there is no MSF relative to the minima of F associated to the cut produced by the topological watershed H. Observe, in particular, that the produced cut is not located on the highest "crests" of the original map F.

An important consequence of Prop. 35 is that border cuts (hence, by Th. 20, watershed cuts) directly inherit all the properties of W-thinnings and topological watersheds proved for vertex-weighted graphs [8–10].

In recent papers [32,38,39], we have studied and proposed solutions to some of the problems encountered by region merging methods which consider frontiers made of vertices as initial segmentations. In particular, we have introduced an adjacency relation on $\mathbb{Z}^n$ which is adapted for region merging. An important property (Prop. 54 in [38]) is that the induced grids, called the perfect fusion grids, are line graphs. If we consider a map which assigns a weight to the vertices of such a grid, then the set of definitions and properties given in this paper are still valid. Thus, the perfect fusion grids constitute an interesting alternative for defining a watershed of an image which is based on vertices and which satisfies the drop of water principle.

7 Illustrations and experimental results

In order to illustrate the notions introduced in this paper, we present two segmentation schemes based on watersheds and relative MSFs. In Sec. 7.1, we derive, from the classical framework of mathematical morphology, a segmentation scheme that allows to automatically segment an image into a predefined number of regions. It consists of the three following steps: (i), computation of a simple function that assigns a weight to the edges of the 4-adjacency graph associated to the image; (ii), filtering of this cost function in order to reduce the number of minima; and (iii) computation of a watershed of the filtered cost function. The second illustration (Sec. 7.2) presents some results of relative MSF, used as a semi-automatic segmentation tool. At last, we provide computation times of several watershed algorithms including the two that are proposed in this paper.

7.1 Segmentation into k regions

In order to illustrate the use of watersheds in practical applications, we adapt a classical scheme of morphological segmentation. We assume that the set V is the domain of a 2-dimensional image, more precisely, of a rectangular subset of $\mathbb{Z}^2$. A grayscale image I is a map from the set of pixels V to a subset of the positive integers. For any $x \in V$, the value $I(x)$ is the intensity at pixel x. We consider the 4-adjacency relation [17] defined by: $\forall x, y \in V$, $\{x, y\} \in E$ iff $|x_1 - y_1| + |x_2 - y_2| = 1$, where $x = (x_1, x_2)$ and $y = (y_1, y_2)$. We consider the map F, from E to $\mathbb{Z}$, defined for any $\{x, y\} \in E$ by $F(\{x, y\}) = |I(x) - I(y)|$. Notice that more elaborated formulations can be used to define the cost function F (see [40, 41] or an adaptation of [42]).

A watershed of F would contain to many catchment basins. Over-segmentation is a well known feature of all grayscale watersheds due to the huge number of local minima. In order to suppress many of the non-significant minima, a classical approach consists of computing morphological closing of the function [43,44]. In particular, attribute filters [45] (area, dynamic, volume) have shown to be successful tools. For this illustration, we adapt a classical attribute filter to the case of edge-weighted graphs.

The intuitive idea of this filter is to progressively “fill in” the minima of the map F that are not "important enough". To make such an idea practicable, it is necessary to quantify the relative importance of a minimum. To this aim, let us define the *area* of a subgraph of G (*e.g.*, a minimum of F) as the number of its vertices. In order to "fill in" a less significant minimum M of F (according to its area), we consider the transformation that consists of increasing by one the altitude of any edge of M. A common issue in image analysis is to segment an image into k regions (where k is a predefined number). To reach this goal thanks to watershed cut, we need a cost function which contains exactly k minima. The map F is thus filtered by iterating the above transformation until F contains k minima (see [46] for an efficient implementation).

In Figs. 9a,b, we present the results which have been obtained on the cameraman image. Here, k is set to 22. In order to evaluate this result, we also use a similar approach settled in the framework of vertex-weighted graph. More precisely, it consists of: (i), computation of a gradient magnitude image: either the Deriche's optimal edge detector [42] in Figs. 9c,d or the morphological gradient (see, for instance, chapter 3.10.1 in [47]) in Figs. 9e,f; (ii), area filtering ($k = 22$) of the gradient; followed by (iii), computation of a watershed by flooding (without dividing line, see [4] or [13]) of the filtered function. Observe, in particular, the quality of the delineation of the man's face in (b) compared to (d) and (f).

7.2 Semi-automatic scheme for image segmentation

Another classical procedure in mathematical morphology consists of using the watershed in an interactive manner. In this procedure, the user "paints", on the image, some markers corresponding to objects that have to be segmented. Actually, the action of "painting" corresponds to the selection of some vertices of the underlying graph. Let M be this set of vertices. From the set M the subgraph M^+ whose vertex set is M and whose edge set is made of the edges of G which have their extremities in M, (*i.e.*, $M^+ = (M, \{\{x, y\} \in E \text{ with } x \in M, y \in M\})$) is extracted. Then, a MSF relative to M^+ is computed. Here, we use a Prim-like minimum spanning tree algorithm [26]. We note that it is possible to efficiently compute minimum spanning trees by an algorithm which consists of a succession of watersheds [48]. Such an algorithm could be also used to produce relative MSFs.

This interactive segmentation procedure is illustrated in Figs. 10a,b and c. For comparison purpose, we also compute the watershed by flooding from markers [4] of the gradient magnitude (the Deriche optimal edge detector [42] in Figs. 10d and morphological gradient in Figs. 10e). We can observe the quality of the delineation in 10c, compared to (d) and (e). See, in particular, the behavior of our approach in low contrasted zones and in the thin parts of the apple.

7.3 Computation times of watershed algorithms

In Fig. 11, computation times (on a conventional personal computer) of several watershed algorithms are plotted for different image sizes. Minima precomputation is included in the execution time of the algorithms that have such a requirement. For each image size, the plotted values correspond to the mean time over ten tested images. Remark that, in accordance with the theoretical study, the two algorithms proposed in this paper are the fastest ones. For example, Flow Cut Algorithm runs five times faster than Meyer's flooding algorithm for images of 2048^2 pixels.

Conclusion

Fig. 12 provides a summary of the main results of this paper. In a unifying framework of edge-weighted graphs, we have shown strong links existing between several paradigms linked to the notion of watershed: topographical paradigms, grayscale transforms paradigms and optimality paradigms. To this aim, we introduced new watershed notions (watershed, basin, border and M-border cuts) and proved the equivalence between watershed cuts, basin cuts, flooding cuts, border cuts, M-border cuts, SPF cuts relative to the minima and MSF cut relative to the minima. Furthermore, we have shown that all these cuts are topological cuts, and thus inherit the mathematical properties of topological watersheds. We proposed two original algorithms (based on grayscale transforms and topographical paradigms) to compute watershed cuts. These two algorithms run in linear time whatever the range of the input function. To our best knowledge, these are the first watershed algorithms satisfying such a property. Furthermore, according to our experiments, they are also the fastest ones. Finally, the defined concepts have been illustrated in image segmentation leading to the conclusion that our approach can be applied for improving the quality of segmentation methods based on watershed.

On the one hand, future works will be focused on hierarchical segmentation schemes based on watersheds (including *geodesic saliency of watershed contours* [49] and *incremental MSFs*) as well as on watershed in weighted simplicial complexes, an image representation adapted to the study of topological properties. On the other hand, we will study a new minimum spanning tree algorithm based on watersheds.

References

[1] J. Maxwell, On hills and dales, Philosophical Magazine 4/40 (1870) 421–427.

[2] C. Jordan, Nouvelles observations sur les lignes de faîtes et de thalweg, Comptes Rendus des Séances de l'Académie des Sciences 75 (1872) 1023–1025.

[3] S. Beucher, C. Lantuéjoul, Use of watersheds in contour detection, in: Procs. of the International Workshop on Image Processing Real-Time Edge and Motion Detection/Estimation, 1979.

[4] F. Meyer, Un algorithme optimal de ligne de partage des eaux, in: Procs. of 8ème Congrès AFCET, Lyon-Villeurbanne, France, 1991, pp. 847–859.

[5] L. Vincent, P. Soille, Watersheds in digital spaces: An efficient algorithm based on immersion simulations, IEEE Trans. Pattern Analysis and Machine Intelligence 13 (6) (1991) 583–598.

[6] S. Beucher, F. Meyer, The morphological approach to segmentation: the watershed transformation, E. Dougherty (Ed.), Mathematical Morphology in Image Processing, Marcel Decker (1993) 443–481.

[7] M. Couprie, G. Bertrand, Topological grayscale watershed transform, in: Procs. of SPIE Vision Geometry V, Vol. 3168, 1997, pp. 136–146.

[8] G. Bertrand, On topological watersheds, Journal of Mathematical Imaging and Vision 22 (2-3) (2005) 217–230.

[9] M. Couprie, L. Najman, G. Bertrand, Quasi-linear algorithms for the topological watershed, Journal of Mathematical Imaging and Vision 22 (2-3) (2005) 231–249.

[10] L. Najman, M. Couprie, G. Bertrand, Watersheds, mosaics and the emergence paradigm, Discrete Applied Mathematics 147 (2-3) (2005) 301–324.

[11] F. Meyer, Minimum spanning forests for morphological segmentation, in: Procs. of the second international conference on Mathematical Morphology and its Applications to Image Processing, 1994, pp. 77–84.

[12] A. X. Falcão, J. Stolfi, R. de Alencar Lotufo, The image foresting transform: theory, algorithm and applications, IEEE Trans. Pattern Analysis and Machine Intelligence 26 (2004) 19–29.

[13] R. Lotufo, A. Falcão, The ordered queue and the optimality of the watershed approaches, in: Procs. of the 5th International Symposium on Mathematical Morphology, 2000, pp. 341–350.

[14] J. K. Udupa, S. Samarsekara, Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation, Graphical Models and Image Processing 58 (1996) 246–261.

[15] P. K. Saha, J. K. Udupa, Relative fuzzy connectedness among multiple objects: Theory, algorithms, and applications in image segmentation, Computer Vision and Image Understanding 82 (2001) 42–56.

[16] R. Diestel, Graph Theory, Graduate Texts in Mathematics, Springer, 1997.

[17] T. Kong, A. Rosenfeld, Digital topology: Introduction and survey, Computer Vision, Graphics, and Image Processing 48 (3) (1989) 357–393.

[18] E. Dahlhaus, D. Johnson, C. Papadimitriou, P. Seymour, M. Yannakakis, Complexity of multiway cuts, in: Procs. of the 24th Annual ACM Symposium on the Theory of Computing, 1992, pp. 241–251.

[19] F. Meyer, Topographic distance and watershed lines, Signal Processing 38 (1) (1993) 113–125.

[20] L. Najman, M. Schmitt, Watershed of a continuous function, Signal Processing 38 (1) (1993) 68–86.

[21] P. Soille, C. Gratin, An efficient algorithm for drainage network extraction on DEMs, Journal of Visual Communication and Image Representation 5 (2) (1994) 181–189.

[22] J. B. T. M. Roerdink, A. Meijster, The watershed transform: Definitions, algorithms and parallelization strategies, Fundamenta Informaticae 41 (1-2) (2001) 187–228.

[23] J. Cousty, G. Bertrand, L. Najman, M. Couprie, The drop of water principle: comparison of classical watershed algorithms, Technical report In preparation.

[24] J. Nesetril, E. Milková, H. Nesetrilová, Otakar Boruvka on minimum spanning tree problem. Translation of both 1926 papers, comments, history, Discrete Mathematics 233 (2001) 3–36.

[25] J. Kruskal, On the shortest spanning tree of a graph and the traveling salesman problem, procs. Amer. Math. Soc. 7 (1956) 48–50.

[26] R. Prim, Shortest connection networks and some generalizations, Bell Syst. Tech. J. 36 (1957) 1389–1401.

[27] R. L. Graham, P. Hell, On the history of the minimum spanning tree problem, Annals of the History of Computing 7 (1985) 43–57.

[28] C. Zahn, Graph-theoretical methods for detecting and descibing gestalt clusters, IEEE Transactions on Computers C-20 (1) (1971) 99–112.

[29] T. H. Cormen, C. Leiserson, R. Rivest, Introduction to algorithms, second edition, MIT Press, 2001.

[30] B. Chazelle, A minimum spanning tree algorithm with inverse-Ackermann type complexity, Journal of the ACM 47 (2000) 1028–1047.

[31] A. N. Moga, T. Viero, M. Gabbouj, M. Nölle, G. Schreiber, H. Burkhardt, Parallel watershed algorithm based on sequential scanning, in: Procs. of IEEE Workshop on Nonlinear Signal and Image Processing, 1995.

[32] J. Cousty, M. Couprie, L. Najman, G. Bertrand, Grayscale watersheds on perfect fusion graphs, in: Procs. of the 11th International Workshop on Combinatorial Image Analysis, Vol. LNCS 4040, 2006, pp. 60–73.

[33] R. Audigier, R. A. Lotufo, Duality between the watershed by image foresting transform and the fuzzy connectedness segmentation approaches, in: Procs. of the XIX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI), 2006, pp. 53–60.

[34] G. Bertrand, On the dynamics, Image and Vision Computing to appear.

[35] A. Rosenfeld, On connectivity properties of grayscale pictures, Pattern Recognition 16 (1983) 47–50.

[36] J. K. Udupa, P. K. Saha, R. A. Lotufo, Relative fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation, IEEE Trans. Pattern Analysis and Machine Intelligence 24 (11) (2002) 1485–1500.

[37] C. Berge, Graphes et hypergraphes, Dunod, 1970.

[38] J. Cousty, G. Bertrand, M. Couprie, L. Najman, Fusion graphs: merging properties and watershed, submitted. Also in technical report IGM2005-04 (2006).
URL http://igm.univ-mlv.fr/LabInfo/rapportsInternes/2005/04.pdf

[39] J. Cousty, G. Bertrand, M. Couprie, L. Najman, Fusion graphs, region merging and watersheds, in: Procs. of the 13th International Conference on Discrete Geometry for Computer Imagery, Vol. LNCS 4245, 2006, pp. 343–354.

[40] P. K. Saha, J. K. Udupa, D. Odhner, Scale-based connected image segmentation: Theory, algorithms and validation, Computer Vision and Image Understanding 77 (2000) 145–174.

[41] Y. Zhuge, J. K. Udupa, P. K. Saha, Vectorial scale-based fuzzy-connected image segmentation, Computer Vision and Image Understanding 101 (2006) 177–193.

[42] R. Deriche, Using Canny's criteria to derive a recursively implemented optimal edge detector, The International Journal of Computer Vision 1 (2) (1987) 167–187.

[43] J. Serra, Image Analysis and Mathematical Morphology, Vol. II: Theoretical Advances, Academic Press, 1988.

[44] J. Serra, L. Vincent, An overview of morphological filtering, Circuits Systems Signal Process 11 (1) (1992) 48–107.

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

[46] L. Najman, M. Couprie, Building the component tree in quasi-linear time, IEEE Trans. Image Processing 15 (11) (2006) 3531–3539.

[47] P. Soille, Morphological Image Analysis, Springer-Verlag, 1999.

[48] J. Cousty, L. Najman, G. Bertrand, M. Couprie, Minimum spanning tree by watershed in preparation.

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

A Proofs

A.1 Proof of Sec. 2

The following lemma is a direct consequence of the definition of a minimum.

Lemma 36 *Let $P \subseteq V, P \neq \emptyset$. If there is no vertex of $M(F)$ in P, then there exists an edge $u = \{x, y\}$ of G such that $x \in P$, $y \in V \setminus P$, and $F(u)$ is less than or equal to the altitude of any vertex in P.*

Proof of Th. 6:
(ii) Suppose that S is a basin cut of F. Let $u = \{x_0, y_0\}$ be any edge in S. There exists $\pi_1 = \langle x_0, \ldots, x_l \rangle$ (resp. $\pi_2 = \langle y_0, \ldots, y_m \rangle$) a path with steepest descent from x_0 (resp. y_0) to $M(F)$. By definition of a cut, x_0 and y_0 are in two distinct connected components of $\overline{S}$. Thus, since $\overline{S}$ is an extension of $M(F)$, x_l and x_m are necessarily in two distinct minima of F. Whenever π_1 (resp. π_2) is not trivial, by definition of a path with steepest descent, $F(\{x_0, x_1\}) = F(x_0)$ (resp. $F(\{y_0, y_1\}) = F(y_0)$). Hence, $F(\{x_0, x_1\}) \leq F(\{x_0, y_0\})$ (resp. $F(\{y_0, y_1\}) \leq F(\{x_0, y_0\})$). Hence, since by definition $\overline{S}$ is an extension of $M(F)$, S is a watershed cut of F.
(ii) Suppose now that S is not a basin cut of F. If $\overline{S}$ is not an extension of $M(F)$, S is not a watershed of F. Suppose now that $\overline{S}$ is an extension of $M(F)$. Thus, there exists a point $x \in V$ such that there is no path with steepest descent in $\overline{S}$ from x to $M(F)$ (otherwise S would be a basin cut of S). Let P be the set of all points of G that can be reached from x by a path with steepest descent in $\overline{S}$. By hypothesis, none of the points in P is a vertex of $M(F)$. We denote by T the set of all edges with minimal altitude among the edges $\{y, z\}$ such that $y \in P$, $z \in V \setminus P$. Let $v = \{y, z\} \in T$ such that $y \in P$. Since none of the vertices of P is a vertex of $M(F)$, from Lem. 36, we can deduce that $F(y) = F(\{y, z\})$. Thus, there is, from x to z, a path in G, with steepest descent for F. Since z is not in P, there is no such path in $\overline{S}$. Thus, $v \in S$ and $T \subseteq S$. Again, let us consider $v = \{y, z\} \in T$. Let $\pi = \langle y_0 = y, \ldots, y_l \rangle$ be any descending path in $\overline{S}$ from y to $M(F)$. If such a path does not exist, then S is not a watershed: the proof is done. Suppose now that such a path exists. There exists $k \in [1, l]$ such that $y_{k-1} \in P$ and $y_k \in V \setminus P$. Since any edge in T is in S and since $\{y_{k-1}, y_k\}$ is in $\overline{S}$, $F(\{y_{k-1}, y_k\}) > F(v)$. Thus, as π is descending, $F(\{y_0, y_1\}) > F(v)$. Thus, the edge v, which belongs to S, does not satisfy the condition for the edges in a watershed: S is not a watershed. $\square$

Before proving the properties of Sec. 3, let us state the following propositions whose proofs are elementary.

Thanks to the construction presented in Sec. 3.2, we can derive, from classical properties of trees, the following properties.

Let $X \subseteq G$, $u \in E(X)$. We write $X \setminus u$ for $(V(X), E(X) \setminus \{u\})$. Let $v \in E \setminus E(X)$. We write $X \cup v$ for the graph $(V(X) \cup v, E(X) \cup \{v\})$.

Lemma 37 *Let X be a subgraph of G and let Y be a spanning forest relative to X. If for any $u \in E(Y) \setminus E(X)$ and $v \in E \setminus E(Y)$ such that $(Y \setminus u) \cup v$ is a spanning forest relative to X, we have $F(u) \leq F(v)$, then Y is a MSF relative to X.*

Lemma 38 *Let X be a subgraph of G and Y be a spanning forest relative to X. If $u = \{x, y\} \in E(Y) \setminus E(X)$, then there exists a unique component of $Y \setminus u$ which does not contain a component of X. Furthermore, either x or y is a vertex of this component.*

Let $\pi = \langle x_0, \ldots, x_l \rangle$ be a path in G. We say that π is a *simple path* if for any two distinct i and j in $[0, l]$, $x_i \neq x_j$. We say that π is an *$\mathcal{M}$-path (for F)* if π is a *simple path*, if x_l is a vertex of $M(F)$ and if none of $x_0, \ldots, x_{l-1}$ is a vertex of $M(F)$. Remark that an $\mathcal{M}$-path does not contain any edge of $M(F)$. Furthermore, it may be seen that if Y is a forest relative $M(F)$, there exists a unique $\mathcal{M}$-path from each vertex of Y.

Proof of Lem. 9:
(i) Suppose that there exists x_0, a vertex of X such that there is no path from x_0 to $M(F)$, with steepest descent for F. We are going to prove that X is not a MSF relative to $M(F)$. Let $\pi = \langle x_0, \ldots, x_l \rangle$ be the unique $\mathcal{M}$-path from x_0 in X. Let $i \in [0, l-1]$ be such that $\langle x_0, \ldots, x_i \rangle$ is a path with steepest descent for F and such that $\langle x_0, \ldots, x_{i+1} \rangle$ is not. We have: $F(x_i) < F(\{x_i, x_{i+1}\})$. Let $Z = X \setminus \{x_i, x_{i+1}\}$. Since $\{x_i, x_{i+1}\}$ is not an edge of $M(F)$, from Lem. 38, there exists a unique connected component of Z, denoted by C, which does not contain a minimum of F. Furthermore, the vertex set of C does not contain any vertex of $M(F)$. Since π is an $\mathcal{M}$-path , hence a simple path, $\langle x_{i+1}, \ldots, x_l \rangle$ is a path in Z and x_l is a vertex of $M(F)$. Thus, x_i is a vertex of C. From Lem. 36, we deduce that there exists $v = \{y, z\} \in E$ such that y is a vertex of C whereas z is not and $F(v) \leq F(x_i)$. Thus, $F(v) < F(\{x_i, x_{i+1}\})$. By definition, we have $V(Z) = V(X) = V$. Hence, it may be seen that $Z \cup v$ is a spanning forest relative to $M(F)$ whose weight is strictly less than the weight of X. Thus, X is not a MSF relative to $M(F)$.
(ii) Suppose that X is not a MSF relative to $M(F)$. We are going to prove

that there exists $x \in V$ such that there is no path with steepest descent in X from x to $M(F)$. By the converse of Lem. 37, there exists $u \in E(X) \setminus E(M(F))$ and $v \in E \setminus E(X)$ such that $(X \setminus u) \cup v$ is a spanning forest relative to $M(F)$ and $F(v) < F(u)$. Let $X' = X \setminus u$. By Lem. 38, there exists a unique connected component of X', denoted by C, which does not contain any minimum of F. Since $X' \cup v$ is an extension of $M(F)$, there exists a unique vertex x in v which is a vertex of C. As $x \in v$, $F(x) \leq F(v)$. Thus, $F(v) < F(u)$ implies $F(x) < F(u)$. Let π be the unique $\mathcal{M}$-path in X from x to $M(F)$. Since C does not contain any minimum of F, we deduce that π passes through u but $F(x) < F(u)$. Hence, π is not a path with steepest descent for F. $\square$

The following lemmas will be used in the proof of Th. 10.

Lemma 39 *Let $S \subseteq E$ be a watershed of F and $Y \subseteq \overline{S}$ be a forest relative to $M(F)$. If $V(Y) \neq V$, then there exists an edge $\{x, y\}$ in $\overline{S}$ outgoing from Y such that either $\langle x, y\rangle$ or $\langle y, x\rangle$ is a path with steepest descent for F. Furthermore, $Y \cup \{x, y\}$ is a forest relative to $M(F)$.*

Proof: Since $V(Y) \neq V$, there exists $x_0 \in V \setminus V(Y)$. Since S is a watershed, by Th. 6, there exists, from x_0 to $M(F)$, a path $\pi = \langle x_0, \ldots, x_l\rangle$ in $\overline{S}$ with steepest descent for F. Since $M(F) \subseteq Y$, there exists $i \in [0, l-1]$ such that $x_i \notin V(Y)$ and $x_{i+1} \in V(Y)$. Thus, $\{x_i, x_{i+1}\}$ is outgoing from Y. Furthermore, by the very definition of a path with steepest descent for F, $\langle x_i, x_{i+1}\rangle$ is a path with steepest descent for F.
Since $x_i \notin V(Y)$, any cycle in $Y \cup \{x_i, x_{i+1}\}$ is also a cycle in Y. Thus, thanks to Rem. 7, $Y \cup \{x_i, x_{i+1}\}$ is a forest relative to Y, hence a forest relative to $M(F)$. $\square$

The following lemma follows straightforwardly from the definition of a path with steepest descent.

Lemma 40 *If $\langle x_0, \ldots, x_l\rangle$ and $\langle x_l, \ldots, x_m\rangle$ are two paths with steepest descent for F, then $\pi = \langle x_0, \ldots, x_m\rangle$ is a path with steepest descent for F.*

Proof of Th. 10:
(i) If S is a cut induced by a MSF relative to $M(F)$, then, by Lem. 9, there exists a path with steepest descent in $\overline{S}$ from each point in V to $M(F)$. Hence, by Th. 6, S is a watershed of F.
(ii) Suppose that S is a watershed of F. Let us consider a sequence of graphs $X_0, \ldots, X_k$ such that:
- $X_0 = M(F)$;
- $X_{i+1} = X_i \cup \{x_i, y_i\}$ where $\{x_i, y_i\}$ is an edge of $\overline{S}$ outgoing from X_i such that $\langle x_i, y_i\rangle$ is a path with steepest descent for F;
- X_k is such that there is no edge $\{x_k, y_k\}$ of $\overline{S}$ outgoing from X_k such that $\langle x_k, y_k\rangle$ is a path with steepest descent for F.
By induction on Lem. 39, X_k is a forest relative to $M(F)$. Furthermore, by

the converse of Lem. 39, $V(X_k) = V$. Thus, X_k is a spanning forest relative to $M(F)$. From Lem. 40, it can be deduced by induction that for any $x \in V$ there exists, from x to $M(F)$, a path in X_k with steepest descent for F. Hence, by Lem. 9, X_k is a MSF relative to $M(F)$. Furthermore, since S is a cut and $X_k \subseteq \overline{S}$, it may be seen that S is the cut induced by X_k. $\square$

A.3 Proofs of Sec. 4

Lemma 41 *Let $H \in \mathcal{F}$. If H is a border thinning of F, then any forest relative to $M(H)$ is a forest relative to $M(F)$.*

Proof: Let u be a border edge for F and let H be the lowering of F at u. We first prove the property for H. Then, Lem. 41 can be easily established by induction. If u is not an edge of $M(H)$ then $M(H) = M(F)$: the proof is done. Suppose now that u is an edge of $M(H)$. Let $u = \{x, y\}$ with $F(x) \geq F(y)$. The fact that u is border for F implies $F(u) = F(x)$ and $F(u) > F(y)$. Thus, u is not an edge of $M(F)$ and x cannot belong to an edge of $M(F)$ (otherwise we would have $F(x) < F(u)$). Therefore, x is not a vertex of $M(F)$. The edge u belongs to S, the edge set of one minimum of H. Since $H(u) = F(y)$ and $F(u) > F(y)$ (by definition of a lowering at a border edge), there is an edge $v \neq u$ which contains y such that $F(v) = H(v) = F(y) = H(u)$. Necessarily v belongs to S. Hence, $S \setminus u \neq \emptyset$ and it may be seen that $S \setminus u$ is exactly the edge set of a minimum of F. Thus y is a vertex of $M(F)$ and $M(H)$ is an extension of $M(F)$. Furthermore, since x is not a vertex of $M(F)$, any cycle in $M(H)$ is also a cycle in $M(F)$. Thus, by Rem. 7, $M(H)$ is a forest relative to $M(F)$ and any forest relative to $M(H)$ is also a forest relative to $M(F)$. $\square$

Lemma 42 *Let H be a border thinning of F.*
(i) For any vertex x of a minimum of H, there exists a path in $M(H)$ from x to $M(F)$ which is a path with steepest descent for F.
(ii) Any $\mathcal{M}$-path (for H), with steepest descent for H is a path with steepest descent for F.

Proof:
Let us first suppose that H is the lowering of F at a border edge u for F.
(i) Let x and y be the two vertices in u. If none of x and y is a vertex of $M(F)$, then $M(F) = M(H)$ and the proof is trivial. Suppose that y is a vertex of $M(F)$. Since u is a border edge, $F(x) = F(u)$. Thus, $\langle x, y \rangle$ is a path in $M(H)$ with steepest descent for F. Let z be any vertex of $M(H)$, $z \neq x$. Necessarily z is also a vertex of $M(F)$. Hence, $\langle z \rangle$ is a path in $M(H)$ from z to $M(F)$ with steepest descent for F.
(ii) The property is verified for any trivial path. Let us consider the case of non-trivial paths. Let $x_0 \in V \setminus V(M(H))$ and let $\pi = \langle x_0, \ldots, x_l \rangle$ be an

$\mathcal{M}$-path (for H) with steepest descent for H. Since x_l is a vertex of $M(H)$ and $\{x_{l-1}, x_l\}$ is not an edge of $M(H)$, by the very definition of a minimum, $H(\{x_{l-1}, x_l\}) > H(x_l)$. Hence, from the definition of a lowering, we deduce that $u \neq \{x_{l-1}, x_l\}$.
Suppose that there exists $i \in [1, l-1]$ such that $u = \{x_{i-1}, x_i\}$. As π is a path with steepest descent for H, $H(x_i) = H(\{x_i, x_{i+1}\})$. By the very definition of a lowering, $\{x_{i-1}, x_i\}$ is the only edge of G whose altitude is different for F and H. Thus, $F(\{x_i, x_{i+1}\}) = H(\{x_i, x_{i+1}\}) = H(x_i)$ and, by definition, $F(x_i) \leq H(x_i)$. Since H is a lowering of F, $H(x_i) \leq F(x_i)$. Hence, $F(x_i) = H(x_i) = F(\{x_i, x_{i+1}\})$. Therefore, since $F(\{x_{i-1}, x_i\}) > H(\{x_{i-1}, x_i\})$, necessarily $F(\{x_{i-1}, x_i\}) > F(x_i)$ and since u is border for F, $F(x_{i-1}) = F(\{x_{i-1}, x_i\})$. Furthermore, for any $v \in E$, $v \neq u$, $F(v) = H(v)$. Thus, in this case, π is a path with steepest descent for F.
Suppose now that for any $i \in [1, l]$, $u \neq \{x_{i-1}, x_i\}$. By definition of a lowering $F(u) > H(u)$, hence, for any $i \in [0, l]$, $F(x_i) = H(x_i)$. Thus π is a path with steepest descent for F.
By induction on (i) and (ii) and thanks to Lem. 40, it may be seen that Lem. 42 holds for any border thinning of F. $\square$

Lemma 43 *The map F is a border kernel if and only if V is the vertex set of $M(F)$.*

Proof:
(i) Suppose that V is not the vertex set of $M(F)$. Then, there exists $x_0 \in V$ which is not a vertex of $M(F)$. Since (V, E) is finite, there exists an $\mathcal{M}$-path $\pi = \langle x_0, \ldots, x_l \rangle$ with steepest descent for F. Thus, $F(x_{l-1}) = F(\{x_{l-1}, x_l\})$. Since π is an $\mathcal{M}$-path, $F(x_l) < F(\{x_{l-1}, x_l\})$. Hence, $\{x_{l-1}, x_l\}$ is a border edge for F, and F is not a border kernel.
(ii) Suppose that F is not a border kernel. There exists $u = \{x, y\}$ which is a border edge for F. Without loss of generality, assume that $F(x) = F(u)$ and $F(y) < F(u)$. There is no minimum of F whose vertex set contains x since $F(x) = F(u)$ and since there exists an edge which contains y whose altitude is strictly less than the altitude of u. Hence, V is not the vertex set of $M(F)$. $\square$

Proof of Prop. 13:
(i) Let X be a MSF relative to $M(H)$ for H. We will prove that for any point x_0 in V, there exists in X a path from x_0 to $M(F)$ which is a path with steepest descent for F. Thus, by Lem. 9, this will establish the first part of Prop. 13. From Lem. 9, it may be seen that there exists in X an $\mathcal{M}$-path (for H), denoted by $\pi = \langle x_0, \ldots, x_l \rangle$, which is a path with steepest descent for H. By Lem. 42.ii, π is a path with steepest descent for F. Since x_l is a vertex of $M(H)$, by Lem. 42.i, there exists in $M(H)$ a path $\pi' = \langle x_l, \ldots, x_m \rangle$ from x_l to $M(F)$ which is a path with steepest descent for F. Since X is an extension of $M(H)$, $M(H) \subseteq X$. Hence, π' is a path in X. Moreover, π is by

construction a path in X. Therefore, $\pi'' = \langle x_0, \ldots, x_m \rangle$, is a path in X. Since both π and π' are paths in X with steepest descent for F, by Lem. 40, π'' is also a path in X with steepest descent for F, which, by construction, is a path from x_0 to $M(F)$.
(ii) Suppose that H is a border kernel of F. From Lem. 43, $V(M(H)) = V$. Then, any MSF relative to $M(H)$ is equal to $M(H)$. Hence, from (i), we prove (ii). $\square$

Proof of Prop. 18
Let $u = \{x, y_0\} \in E$, with x being a vertex of $M(F)$, be a flooding edge for F. By the very definition of a minimum, we have $F(u) > F(x)$. Let $\pi = \langle y_0, \ldots, y_l \rangle$ be any $\mathcal{M}$-path with steepest descent for F. It may be seen that $F(\{y_{l-1}, y_l\}) \leq F(y_0)$. Since π is an $\mathcal{M}$-path , $\{y_{l-1}, y_l\}$ is outgoing from $M(F)$. By hypothesis, $F(u) \leq F(\{y_{l-1}, y_l\})$. Thus, $F(u) \leq F(y_0)$ and since $y_0 \in u$, necessarily $F(u) = F(y_0)$. Hence, u is a border edge for F. $\square$

The following lemma is used to prove Lem. 20. The proof is similar to the one of Lem. 43 and, thus, omitted.

Lemma 44 *The map F is an M-border (resp. flooding) kernel if and only if V is the vertex set of $M(F)$.*

Thanks to the construction presented in Sec. 3.2, the following lemma can be derived from basic results on minimum spanning trees (see, for instance, Th. 23.1, p. 563, in [29]).

Lemma 45 *Let X be a subgraph of G, let Y be a MSF relative to X, and let $Z \subseteq Y$ be a forest relative to X such that $Z \neq Y$. Let u be an edge of minimal altitude among all the edges of Y outgoing from Z. Then, the altitude of any edge of G outgoing from Z is greater than or equal to $F(u)$.*

Proof of Lem. 20:
$(i) \implies (ii)$: Let H be a flooding kernel of F and let $X = M(H)$. By Prop. 18, H is a border thinning of F. Consequently to Lem. 44, V is the vertex set of $M(H)$ and, again by Lem. 44, H is an M-border kernel of F. $(ii) \implies (iii)$: Let H be an M-border kernel of F and let $X = M(H)$. Trivially H is a border thinning of F. By Lem. 44, V is the vertex set of $M(H)$. Thus, by Lem. 43, H is a border kernel of F.
$(iii) \implies (iv)$: Prop. 13.
$(iv) \implies (i)$: Let X be a MSF relative to $M(F)$ and let us consider a sequence of graphs $X_0, \ldots, X_k$ such that:
- $X_0 = M(F)$;
- for any $i \in [1, k]$, $X_i = X_{i-1} \cup u_i$ where u_i is an edge of minimal altitude (for F) among all the edges of X outgoing from X_{i-1}; and
- V is the vertex set of X_k.
It may be seen that such a sequence always exists. Consider also the associated

sequence of maps $F_0, \ldots F_k$ such that $F_0 = F$ and for any $i \in [1,k]$, F_i is the lowering of F_{i-1} at u_i.
We will proceed by induction to establish, for any $i \in [1,k]$, the following proposition:
$(\mathcal{P}_i)$: F_i is a flooding of F such that $X_i = M(F_i)$.
Let $i \in [1,k]$ and suppose that $(\mathcal{P}_{i-1})$ holds. By Prop. 18 and Lem. 41, $(\mathcal{P}_{i-1})$ implies that X_{i-1} is a forest relative to $M(F)$. Therefore, it follows from Lem. 45, that the altitude (for F) of any edge of G outgoing from X_{i-1} is greater than or equal to $F(u_i)$. By construction of F_{i-1}, we have $F(v) = F_{i-1}(v)$ for any edge v outgoing from X_{i-1}. Thus, u_i is an edge with minimal altitude (for F_{i-1}) among all the edges outgoing from X_{i-1}. Furthermore, thanks to $(\mathcal{P}_{i-1})$, $X_{i-1} = M(F_{i-1})$. Hence, u_i is a flooding edge for F_{i-1}, and it follows straightforwardly that F_i is a flooding of F_{i-1}. Moreover, by $(\mathcal{P}_{i-1})$, F_i is a flooding of F. Consequently to the definition of a lowering at a flooding edge, $M(F_i) = M(F_{i-1}) \cup u_i$. Hence, $M(F_i) = X_{i-1} \cup u_i = X_i$, which completes the proof of $(\mathcal{P}_i)$.
Since $(\mathcal{P}_0)$ is trivially verified, by induction, $\mathcal{P}_k$ is established. Therefore, by Prop. 18 and Lem. 41, $M(F_k) = X_k$ is a forest relative to $M(F)$. Since $V(X_k) = V$, since $X_k \subseteq X$ (by construction) and since X is a forest relative to $M(F)$, by the definition of a spanning forest, we have necessarily $X_k = X$. By Lem. 44, F_k is a flooding kernel. Hence, by $(\mathcal{P}_k)$, $X = X_k$ is the min-graph of F_k, a flooding kernel of F. $\square$

A.4 Proofs of Sec. 6

Proof of Th. 29: Suppose that Y is a MSF relative to X. Suppose also that there exist A and B, two components of X such that $F(A,B) \neq F(A',B')$, where A' and B' are the two components of Y such that $A \subseteq A'$ and $B \subseteq B'$. Since $\Pi(A,B) \subseteq \Pi(A',B')$, $F(A,B) > F(A',B')$. Let $\pi = \langle x_k, \ldots, x_l \rangle$ be a path from A' to B' such that $F(\pi) = F(A',B')$ and such that x_k (resp. x_l) is the only vertex of A' (resp. B') in π. Notice that $\{x_k, x_{k+1}\}$ and $\{x_{l-1}, x_l\}$ are not edges of Y. Let $\pi_A = \langle x_0, \ldots, x_k \rangle$ (resp. $\pi_B = \langle x_l, \ldots, x_m \rangle$) be a simple path in A' (resp. B'), such that x_0 (resp. x_m) is the only point of π_A (resp. π_B) which is a point of A (resp. B). Since $\pi' = \langle x_0, \ldots, x_m \rangle$ is a path from A to B, $F(\pi') \geq F(A,B)$. Thus, since $F(\pi) < F(A,B)$, we have either $F(\pi_A) \geq F(A,B)$ or $F(\pi_B) \geq F(A,B)$. Without loss of generality, assume that $F(\pi_A) \geq F(A,B)$. Let u be any edge of π_A such that $F(u) = F(\pi_A)$. Since $F(\pi) < F(\pi_A)$, $F(u) > F(\{x_k, x_{k+1}\})$. Since π_A is a simple path in A', since x_0 is the only point of π_A which is in A, and since $\{x_k, x_{k+1}\}$ is not in Y, it may be seen that $(Y \setminus u) \cup \{x_k, x_{k+1}\}$ is a spanning forest relative to X. Since $F(u) > F(\{x_k, x_{k+1}\})$, $(Y \setminus u) \cup \{x_k, x_{k+1}\}$ has a cost strictly less than Y. Thus, Y is not a MSF relative to X, a contradiction. $\square$

Proof of Prop. 30: Suppose that Y is a MSF relative to X which is not a shortest-path spanning forest relative to X. There exists $x_0 \in V(Y)$ such that for any path π in Y from x_0 to X, we have $F(\pi) > F(x_0, X)$. Let $\pi = \langle x_0, \dots, x_l \rangle$ be any such path and suppose, without loss of generality, that π is a simple path. Let $i \in [0, l-1]$ be such that $F(\{x_i, x_{i+1}\}) = F(\pi)$ and let $u = \{x_i, x_{i+1}\}$. We denote by C the connected component of $Y \setminus u$ such that $x_0 \in V(C)$. Since π is a simple path, from Lem. 38, we deduce that C is the unique connected component of $Y \setminus u$ which does not contain a connected component of X. Let $\pi' = \{y_0 = x_0, \dots, y_m\}$ be a path in G from x_0 to X such that $F(\pi') = F(x_0, X)$. Let $j \in [0, m-1]$ be such that $y_j \in V(C)$ whereas $y_{j+1} \notin V(C)$. Let $v = \{y_j, y_{j+1}\}$. Thus, $(Y \setminus u) \cup v$ is a spanning forest relative to X. Necessarily, $F(v) \leq F(\pi')$. Hence, since $F(\pi') = F(x_0, X)$ and $F(\pi) > F(x_0, X)$, $F(v) < F(\pi)$ and $F(v) < F(\{x_i, x_{i+1}\})$. Thus, from the two previous observations, we deduce that Y is not a MSF relative to X, a contradiction. $\square$

Since G is a finite graph, for any $x \in V$ there exists a path π with steepest descent for F from x to $M(F)$. Then, it may be seen that $F(\pi) = F(x) = F(x, M(F))$.

Proof of Prop. 31:
(i) Suppose that X is a spanning forest relative to $M(F)$ which is not a MSF relative to $M(F)$. From Lem. 9, there exists a vertex $x \in V$ such that none of the paths in X from x to $M(F)$ is with steepest descent for F. Let P be the set of all points that can be reached from x by a path in X with steepest descent for F. Let y_0 be the vertex of P with minimal altitude. By hypothesis, y_0 is not a vertex of $M(F)$. Let $\pi = \langle y_0, \dots, y_l \rangle$ be the unique $\mathcal{M}$-path ,in X, from y_0 to $M(F)$. Let $i \in [0, l-1]$ be the lowest index such that $y_i \in P$ and $y_{i+1} \in V \setminus P$. If $F(\{y_i, y_{i+1}\}) = F(y_i)$, then there exists $j \in [0, i-1]$ such that $F(y_j) < F(\{y_j, y_{j+1}\})$ (otherwise y_{i+1} would belong to P) and thus, $F(\{y_j, y_{j+1}\}) > F(y_0)$ (since $F(y_j) \geq F(y_0)$ by definition of y_0, i and j). If $F(\{y_i, y_{i+1}\}) > F(y_i)$, then $F(y_0) < F(\{y_i, y_{i+1}\})$ since $F(y_0) \leq F(y_i)$. In both cases, $F(\pi) > F(y_0)$. From the remark stated above this proof, we have $F(\pi) > F(y_0, M(F))$, hence, X is not a shortest path forest relative to $M(F)$.
(ii) a direct consequence of Prop. 30. $\square$

Proof of Prop. 35: Let $u = \{x, y\} \in E$ be a border edge for F such that $F(u) = F(x) = k$. We will prove that the lowering of F at u is a W-thinning of F, hence, by induction, this will establish Prop. 35. From the definition of a border edge, $F(y) < k$. Thus, there exists a set of edges $S \subseteq E$, such that $S = \{v_i = \{y, y_i\} \in E \mid y_i \neq x \text{ and } F(v_i) < k\}$. Since any element in S contains y, all the edges in S are in the same component of $\overline{F}[k]$. Since $F(x) = k$, none of the edges $v_j = \{x, z_j\} \in E$ with $z_j \neq y$, is in $\overline{F}[k]$. Thus, u is adjacent to exactly one component of $\overline{F}[k]$. Hence, u is W-destructible for F and the map obtained by lowering the value of u by one is a W-thinning of F. By iterating the same arguments, it may be seen that u can be lowered down to $F(y)$. In other words, the lowering of F at u is a W-thinning of F. $\square$

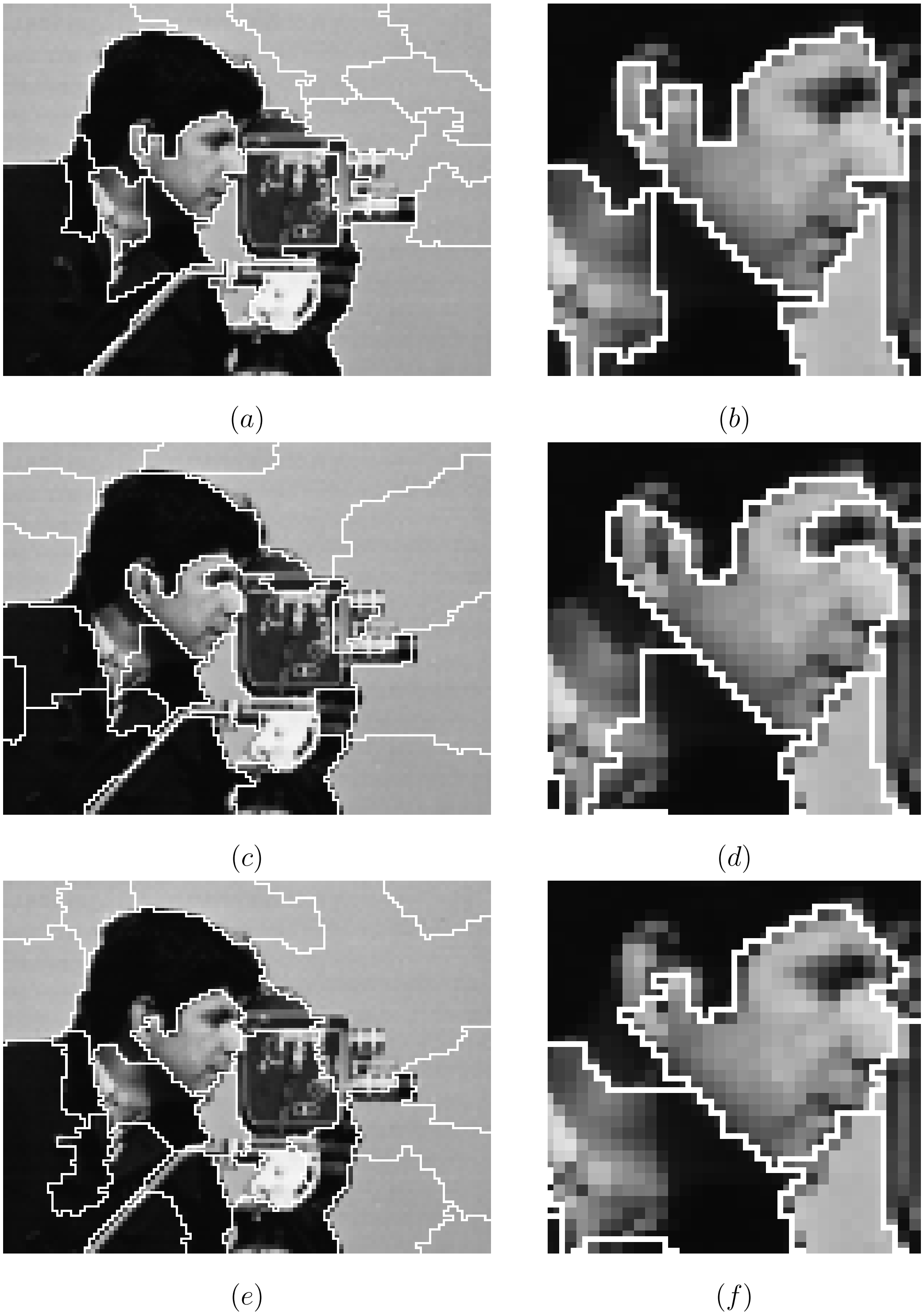


Fig. 9. Results obtained by applying a grayscale watershed on a filtered map. (a, b) A watershed cut ($k = 22$) superimposed in white to the original image I; (c, d) a watershed by flooding of the filtered ($k = 22$) Deriche optimal edge detector; and (e, f) a watershed by flooding of a filtered ($k = 22$) morphological gradient. In each image, the image resolution is doubled in order to superimpose the resulting contours.

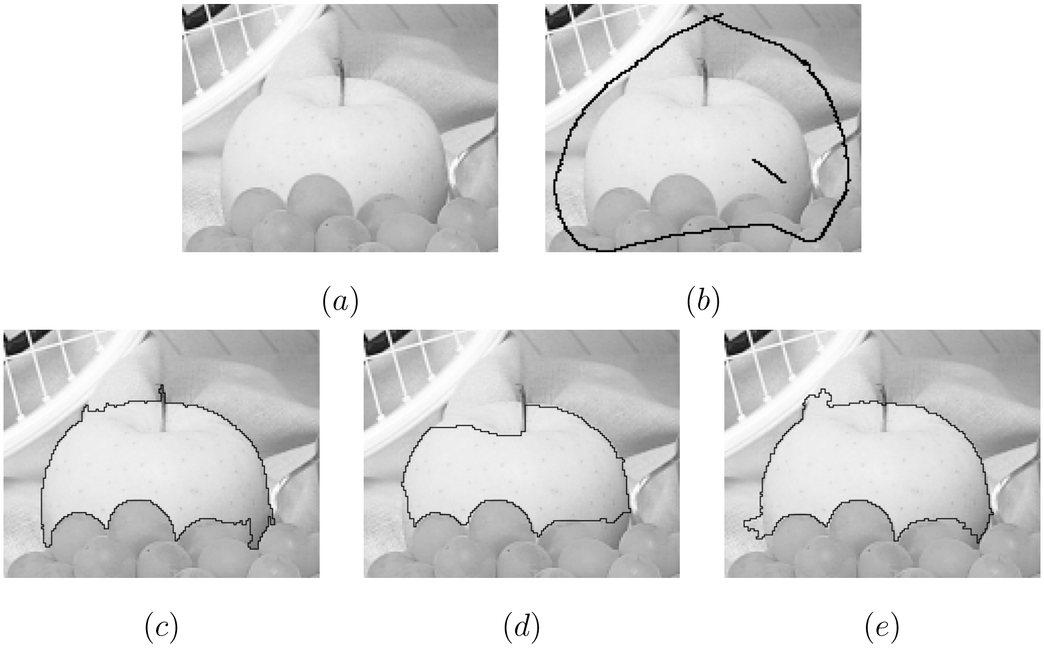


Fig. 10. Comparison of different watersheds from markers. (a) Original image; (b) the markers are superimposed in black. In the second row the resulting watersheds are superimposed in black to the original image. (c): Relative MSF; (d): watershed by flooding of the Deriche optimal edge detector; (e): watershed by flooding of a morphological gradient of the image.

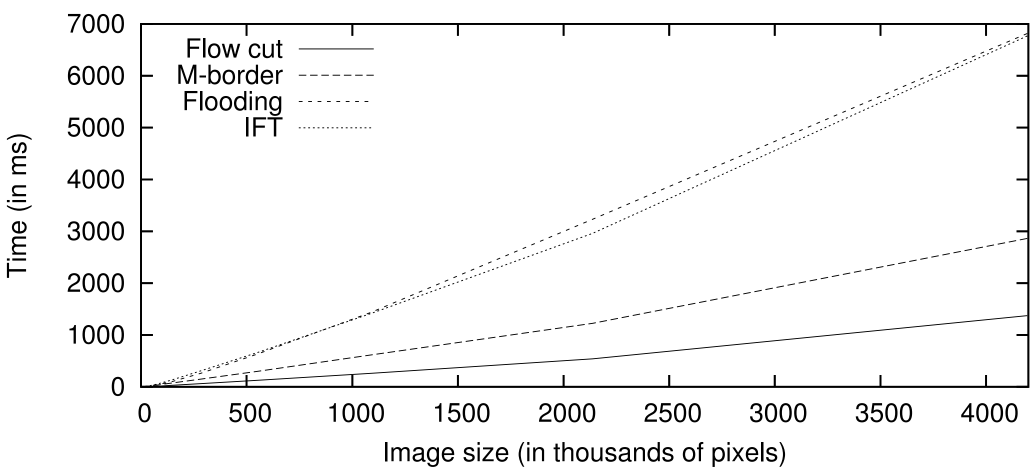


Fig. 11. Computation times of four different watershed algorithms: watershed by flooding (without dividing line) [4], watershed by Image Foresting Transform [13] and the two algorithms (M-border Kernel and flow cut) proposed in this paper

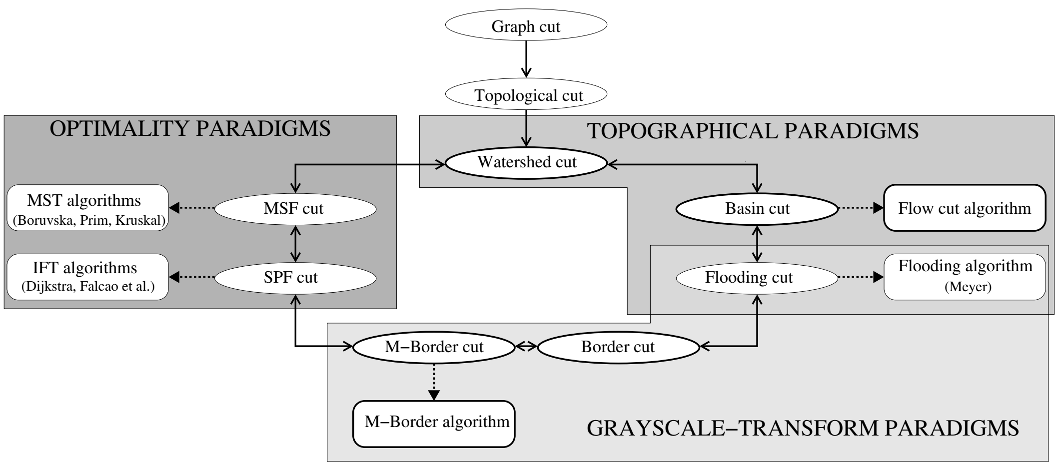


Fig. 12. Summary of the properties on cuts for the regional minima of a map. In the figure, $N \leftarrow N'$ means that the notion N is a particular case of the notion N', hence, $N \leftrightarrow N'$ means that the notions N and N' are equivalent; $A \dashleftarrow N$ means that the notion N can be computed thanks to algorithm A. The new notions and algorithms introduced in this paper are highlighted in bold.