Algorithms for the Topological Watershed

Michel Couprie, Laurent Najman, and Gilles Bertrand

Laboratoire A2SI, Groupe ESIEE,
BP99, 93162 Noisy-le-Grand Cedex France
IGM, Unité Mixte de Recherche CNRS-UMLV-ESIEE UMR 8049
{m.couprie, l.najman, g.bertrand}@esiee.fr

Abstract. The watershed transformation is an efficient tool for segmenting grayscale images. An original approach to the watershed [1, 4] consists in modifying the original image by lowering some points until stability while preserving some topological properties, namely, the connectivity of each lower cross-section. Such a transformation (and its result) is called a topological watershed. In this paper, we propose quasi-linear algorithms for computing topological watersheds. These algorithms are proved to give correct results with respect to the definitions, and their time complexity is analyzed.

1 Introduction

The watershed transformation was introduced as a tool for segmenting grayscale images by S. Beucher and C. Lantuéjoul [2] in the late 70's, and is now used as a fundamental step in many powerful segmentation procedures. The most popular presentation of the watershed is based on a flooding paradigm. Let us consider a grayscale image as a topographical relief: the gray level of a pixel becomes the altitude of a point, the basins and valleys of the relief correspond to the dark areas, whereas the mountains and crest lines correspond to the light areas. Let us imagine the surface of this relief being immersed in still water, with holes pierced in local minima. Water fills up basins starting at these local minima, and dams are built at points where waters coming from different basins would meet. As a result, the surface is partitioned into regions or basins which are separated by dams, called watershed lines. Efficient watershed algorithms based on immersion simulation were proposed by L. Vincent, P. Soille [12] and F. Meyer [6, 3] in the early 90's.

A different approach to watersheds, originally proposed by G. Bertrand and M. Couprie [4], is developed in [1]. In this approach, we consider a transformation called topological watershed, which modifies a map (*e.g.* a grayscale image) while preserving some topological properties, namely, the connectivity of each lower cross-section. It is proved in [1] that, among other properties, topological watersheds satisfy a "constrast preservation" property which is, in general, not satisfied by the most popular watershed algorithms [8].

In this paper, we study algorithms to compute topological watersheds. These algorithms are proved [5] to give correct results with respect to the definition, and their time complexity is analyzed.

2 Topological Notions for Weighted Graphs

Let E be a finite set, we denote by $\mathcal{P}(E)$ the set of all subsets of E. Throughout this paper, Γ will denote a binary relation on E (thus, $\Gamma \subseteq E \times E$), which is reflexive (for all p in E, $(p,p) \in \Gamma$) and symmetric (for all p,q in E, $(q,p) \in \Gamma$ whenever $(p,q) \in \Gamma$). We say that the pair (E,Γ) is a *graph*, each element of E is called a *vertex* or a *point*. We will also denote by Γ the map from E into $\mathcal{P}(E)$ such that, for any p in E, $\Gamma(p) = \{q \in E; (p,q) \in \Gamma\}$. For any point p, the set $\Gamma(p)$ is called the *neighborhood of* p. If $q \in \Gamma(p)$ then we say that p and q are *adjacent* or that q is a *neighbor of* p. If $X \subseteq E$ and q is adjacent to p for some $p \in X$, we say that q *is adjacent to* X.

For applications to digital image processing, assume that E is a finite subset of $\mathbb{Z}^n$ $(n = 2,3)$, where $\mathbb{Z}$ denotes the set of integers. A subset X of E represents the "object", its complementary $\overline{X} = E \setminus X$ represents the "background", and Γ corresponds to an adjacency relation between points of E. In $\mathbb{Z}^2$, Γ may be one of the usual adjacency relations, for example the 4-adjacency or the 8-adjacency in the square grid. In the sequel, the 4-adjacency is assumed.

Let (E,Γ) be a graph, let $X \subseteq E$, and let $p_0, p_k \in X$. A *path from* p_0 *to* p_k *in* X is an ordered family $(p_0, p_1, \ldots, p_k)$ of points of X such that $p_{i+1} \in \Gamma(p_i)$, with $i = 0 \ldots k-1$. Let $p,q \in X$, we say that p *and* q *are linked for* X if there exists a path from p to q in X. We say that X *is connected* if any p and q in X are linked for X. We say that a subset Y of E is a *(connected) component of* X if $Y \subseteq X$, Y is connected, and Y is maximal for these two properties. In the sequel of the article, we will assume that E is connected.

We are interested in transformations that preserve the number of connected components of the background. For this purpose, we introduce the notion of W-simple point (where Wstands for watershed) in a graph. Intuitively, a point of X is W-simple if it may be removed from X while preserving the number of connected components of $\overline{X}$.

Definition 1. Let $X \subseteq E$, let $p \in X$. We say that:
- p is *a border point (for* X*)* if p is adjacent to $\overline{X}$;
- p is *an inner point (for* X*)* if p is not a border point for X;
- p is *separating (for* X*)* if p is adjacent to at least two components of $\overline{X}$;
- p is *W-simple (for* X*)* if p is adjacent to exactly one component of $\overline{X}$.

Notice that a point which is not W-simple is either an inner point or a separating point. In Fig. 2, the points of the set X are represented by "1"s. The points which are W-simple are circled. It may be easily seen that one cannot locally decide whether a point is W-simple or not. Consider the points p and q in the third row: their neighborhoods are alike, yet p is W-simple (it is adjacent to exactly one connected component of $\overline{X}$), and q is not, since it is adjacent to two different connected components of $\overline{X}$.

Now, we extend this notion to a weighted graph (E,Γ,F), where F is a map from E to $\mathbb{Z}$. A weighted graph is a model for a digital grayscale image; for any point $p \in E$, the value $F(p)$ represents the gray level of p. We denote by $\mathcal{F}(E)$ the set composed of all maps from E to $\mathbb{Z}$.

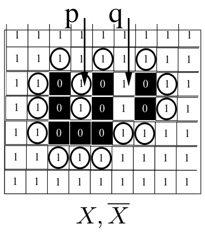


Fig. 1. A set X (the 1's) and its complement $\overline{X}$ (the 0's). All the W-simple points are circled

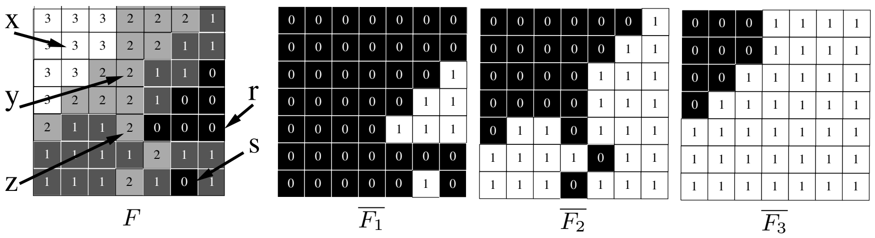


Fig. 2. A grayscale image F and its lower sections $\overline{F_1}$, $\overline{F_2}$ and $\overline{F_3}$ (in white). The points x, r, s are inner points, y is W-destructible (with lowest value 1), and z is separating

Definition 2. Let $F \in \mathcal{F}(E)$, let $k \in \mathbb{Z}$.
We denote by F_k the set $\{p \in E; F(p) \geq k\}$; F_k is called an *upper section of* F, and its complementary $\overline{F_k}$ is called a *lower section of* F.
A component c of $\overline{F_k}$ is called a *(regional) minimum for* F if $c \cap \overline{F_{k-1}} = \emptyset$.
We denote by $\Gamma^-(p, F)$ the *set of lower neighbors of the point* p *for the map* F, that is, $\Gamma^-(p, F) = \{q \in \Gamma(p); F(q) < F(p)\}$. When no confusion may occur, we write $\Gamma^-(p)$ instead of $\Gamma^-(p, F)$.

Fig. 2 shows a grayscale image F and three lower sections of F: $\overline{F_2}$ is made of two components (in white), and $\overline{F_3}$ is made of one component. The set $\overline{F_1}$ is made of two components which are both minima of F.

Definition 3. Let $F \in \mathcal{F}(E)$, let $p \in E$, let $k = F(p)$.
We say that p is *a border point (for F)* if p is a border point for F_k.
We say that p is *an inner point (for F)* if p is an inner point for F_k.
We say that p is *separating (for F)* if p is separating for F_k.
The point p is *W-destructible (for F)* if p is W-simple for F_k. Let $v \in \mathbb{Z}$, $v < F(p)$, the point p is *W-destructible with lowest value* v *(for F)* if for any h such that $v < h \leq F(p)$, p is W-simple for F_h, and if p is not W-simple for F_v.

In other words, the point p is W-destructible for F if and only if p is a border point for F (*i.e.*, $\Gamma^-(p) \neq \emptyset$) and all the points in $\Gamma^-(p)$ belong to the same connected component of $\overline{F_k}$, with $k = F(p)$. In Fig. 2, the points x, r, s are inner points, y is W-destructible (with lowest value 1), and z is separating.

Let $F \in \mathcal{F}(E)$, let $p \in E$, let $v \in \mathbb{Z}$ such that $v < F(p)$, we denote by $[F \backslash p \downarrow v]$ the element of $\mathcal{F}(E)$ such that $[F \setminus p \downarrow v](p) = v$ and $[F \setminus p \downarrow v](q) = F(q)$ for all $q \in E \setminus \{p\}$. Informally, it means that the only difference between the map F and the map $[F \setminus p \downarrow v]$, is that the point p has been lowered down to the value v. We also write $[F \setminus p] = [F \setminus p \downarrow v]$ when $v = F(p) - 1$.

If we consider $F' = [F \setminus p \downarrow v]$, it may be easily seen that for any h in $\mathbb{Z}$, the number of connected components of $\overline{F'_h}$ equals the number of connected components of $\overline{F_h}$. That is to say, the value of a W-destructible point may be lowered by one or down to its lowest value without changing the number of connected components of any lower section of F.

Definition 4. Let $F \in \mathcal{F}(E)$. We say that $G \in \mathcal{F}(E)$ is *a W-thinning of* F if
i) G = F, or if
ii) there exists a map H which is a W-thinning of F and there exists a W-destructible point p for H such that $G = [H \setminus p]$.
We say that G is a *(topological) watershed* of F if G is a W-thinning of F and if there is no W-destructible point for G.

Let $F \in \mathcal{F}(E)$, let $p \in E$, let $v \in \mathbb{Z}$. It may be easily seen that, if p is W-destructible with lowest value v, then $[F \setminus p \downarrow v]$ is a W-thinning of F and p is not W-destructible for $[F \setminus p \downarrow v]$; and that the converse is also true.

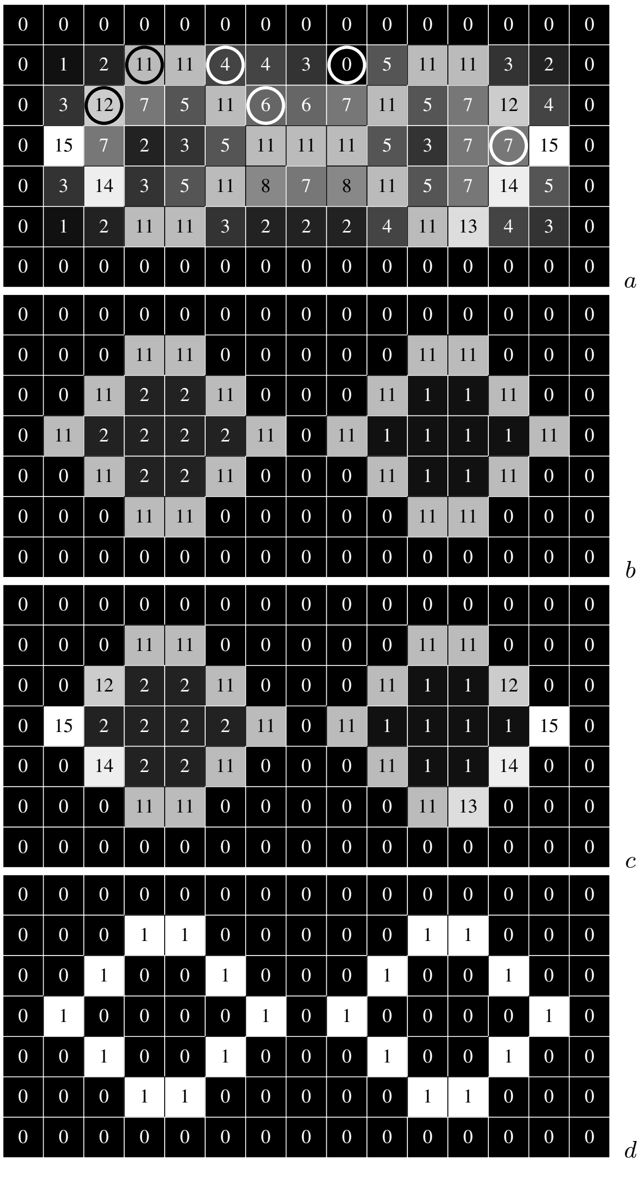
In other words, one can obtain a W-thinning of a map F by iteratively selecting a W-destructible point and lowering it by one, or directly down to its lowest value. If this process is repeated until stability, one obtains a topological watershed of F. Notice that the choice of the W-destructible point is not necessarily unique at each step, thus, in general, there may exist several topological watersheds for the same map.

In Fig. 3, we present an image 3a and a topological watershed 3b of 3a. Note that in 3b, the minima of 3a have been spread and are now separated from each other by a “thin line”; nevertheless, their number and values have been preserved. Fig. 3c shows a W-thinning of 3a which is not a topological watershed of 3a (there are still some W-destructible points).

Let us consider a point $p \in E$ which is not W-destructible for $F \in \mathcal{F}(E)$. Three cases may be distinguished. From Def. 3, such a point is either an inner point or a separating point for F. Furthermore, if p is an inner point, then either p belongs to a minimum of F or not.

On the other hand, if p is W-destructible for F, then p is not W-destructible for $[F \backslash p \downarrow v]$ where v is the lowest value of p. Again, we can distinguish the same three possibilities for the status of p with respect to $[F \setminus p \downarrow v]$. The following definition formalizes these observations (S stands for separating, M for minimum and P for plateau).

Definition 5. Let $F \in \mathcal{F}(E)$, let $p \in E$, p not W-destructible for F.
We say that p is an *S-point (for F)* if p is separating for F.
We say that p is an *M-point (for F)* if p belongs to a minimum of F.
We say that p is an *P-point (for F)* if p is an inner point for F which does not belong to a minimum of F.



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	11	11	4	4	3	0	5	11	11	3	2	0
0	3	12	7	5	11	6	6	7	11	5	7	12	4	0
0	15	7	2	3	5	11	11	11	5	3	7	7	15	0
0	3	14	3	5	11	8	7	8	11	5	7	14	5	0
0	1	2	11	11	3	2	2	2	4	11	13	4	3	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	11	11	0	0	0	0	0	11	11	0	0	0
0	0	11	2	2	11	0	0	0	11	1	1	11	0	0
0	11	2	2	2	2	11	0	11	1	1	1	1	11	0
0	0	11	2	2	11	0	0	0	11	1	1	11	0	0
0	0	0	11	11	0	0	0	0	0	11	11	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	11	11	0	0	0	0	0	11	11	0	0	0
0	0	12	2	2	11	0	0	0	11	1	1	12	0	0
0	15	2	2	2	2	11	0	11	1	1	1	1	15	0
0	0	14	2	2	11	0	0	0	11	1	1	14	0	0
0	0	0	11	11	0	0	0	0	0	11	13	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	1	1	0	0	0
0	0	1	0	0	1	0	0	0	1	0	0	1	0	0
0	1	0	0	0	0	1	0	1	0	0	0	0	1	0
0	0	1	0	0	1	0	0	0	1	0	0	1	0	0
0	0	0	1	1	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 3. a: original image; b: a topological watershed of a; c: a W-thinning of a which is also an M-watershed of a (see Sec. 3); d: a W-crest of a (see Sec. 3). In a, we have circled six points which have different types (see Def. 5). From left to right: $\tilde{\text{S}}$-point (12), S-point (11), $\tilde{\text{M}}$-point (4), $\tilde{\text{P}}$-point (6), M-point (0), P-point (7).

Let q be a point which is W-destructible for F, let v be its lowest value. We say that q is an *$\tilde{S}$-point (for F)* if q is an S-point for $[F \setminus q \downarrow v]$.

We say that q is an *$\tilde{M}$-point (for F)* if q is an M-point for $[F \setminus q \downarrow v]$.
We say that q is a *$\tilde{P}$-point (for F)* if q is a P-point for $[F \setminus q \downarrow v]$.

In Fig. 3a, we have circled six points which are representative of all the possible types.

The components of the lower sections of a map may be organized, thanks to the inclusion relation, in a tree structure that we call *component tree* (see the bibliography of [7] for a list of references). In [5], we propose and prove a characterization of the W-destructible points, which may be checked locally and efficiently implemented thanks the component tree. In [7], we introduce a new algorithm, using Tarjan's union-find procedure [10], to build the component tree of any weighted graph in quasi-linear time, that is, in $O(n \times \alpha(n))$ where n is the size of the graph (number of vertices + number of arcs) and $\alpha(n)$ is a function which grows extremely slowly with n (we have $\alpha(10^{80}) \approx 4$). For a precise definition of α, see [10].

Furthermore, for applications to digital image processing, where each point has a fixed (and small) number of neighbors, we can consider that testing whether a point is W-destructible and computing its lowest value can be done in constant time thanks to the component tree.

3 M-Thinning and Binary Watershed Algorithm

The outline of a topological watershed algorithm is the following:

```
Repeat Until Stability
    Select a W-destructible point $p$, using a certain criterion
    Lower the value of $p$
```

It can be seen that, even if a W-destructible point is lowered down to its lowest value, it may again become W-destructible in further steps of the W-thinning process, due to the lowering of some of its neighbors. For example, the point at level 6 circled in white in Fig. 3a is W-destructible with lowest value 4. If we lower this point down to 4, we will have to lower it again, after the lowering of its neighbor at level 4 down to 3 or 0.

In order to ensure a linear complexity, we must avoid multiple selections of the same point during the execution of the algorithm. The following properties provide selection criteria which guarantee that a point lowered once will never be W-destructible again during the W-thinning process.

The first criterion concerns points which may be lowered down to the value of a neighbor which belongs to a minimum (*i.e.*, $\tilde{\text{M}}$-points). If an $\tilde{\text{M}}$-point is lowered down to its lowest value, then we say that the point is *M-lowered*. The aim of theorem 1 is to show that, if $\tilde{\text{M}}$-points are sequentially selected and M-lowered, and if we continue this process until stability, giving a result G, then it is not possible that a W-thinning of G contains any $\tilde{\text{M}}$-point. Since, obviously, a point which has been M-lowered will never be considered again in a W-thinning algorithm, we obtain an "M-thinning algorithm" which considers each point at most once, and produces a result in which the minima cannot be extended by further W-thinning.

Definition 6. Let $F, G \in \mathcal{F}(E)$, we say that G is an *M-thinning of F* if $G = F$ or if G can be obtained from F by sequentially M-lowering some $\tilde{\mathrm{M}}$-points. We say that G is an *M-watershed of F* if G is a M-thinning of F and has no $\tilde{\mathrm{M}}$-point.

Theorem 1. *Let $F \in \mathcal{F}(E)$, let G be an M-watershed of F. Any W-thinning of G has exactly the same minima as G.*

See [5] for a proof. Let F be a map and let G be a topological watershed of F, the set of points which do not belong to any regional minimum of G is called a *W-crest* of F (see Fig. 3d). A W-crest of F corresponds to a "binary watershed" of F. A corollary of this theorem is that the set of points which do not belong to any minimum of an M-watershed of F is always a W-crest of F. Thus, we can compute a W-crest (or binary watershed) by only lowering $\tilde{\mathrm{M}}$-points. In Fig. 3c, we see an M-watershed of 3a.

In the following algorithm, we introduce a priority function μ which is used to select the next $\tilde{\mathrm{M}}$-point. The priority function μ associates to each point p a positive integer $\mu(p)$, called the priority of p. This function is used for the management of a priority queue, a data structure which allows one to perform, on a set of points, an arbitrary sequence of the two following operations (L denotes a priority queue and p a point):
AddPrioQueue$(L, p, \mu(p))$: store p with the priority $\mu(p)$ into the queue L;
ExtractPrioQueue(L): remove and return a point which has the minimal priority value among those stored in L (if several points fulfill this condition, an arbitrary choice is made).

The choice and the interest of the priority function will be discussed afterwards, but notice that whatever the chosen priority function (for example a constant function), the output of the procedure will always be an M-watershed of the input.

Given a map F and a point p, the procedure call **M-destructible**(F, p) (resp. **W-destructible**(F, p)) returns in constant time (see end of Sec. 2) the lowest value for p if p is an $\tilde{\mathrm{M}}$-point (resp. a W-destructible point), or ∞ otherwise.

```
Procedure M-watershed (Input $F$, $\mu$ ; Output $F$)
    $L \leftarrow$ EmptyPrioQueue
    For All $p \in E$ such that M-destructible($F$, $p$) $\neq \infty$ Do
        AddPrioQueue($L, p, \mu(p)$) ; mark $p$
    While $L \neq$ EmptyPrioQueue Do
        $p \leftarrow$ ExtractPrioQueue($L$) ; unmark $p$
        If M-destructible($F$, $p$) $\neq \infty$ Then
            $F(p) \leftarrow$ M-destructible($F$, $p$)
            For All $q \in \Gamma(p)$, $q \neq p$, $q$ not marked Do
                If M-destructible($F$, $q$) $\neq \infty$ Then
                    AddPrioQueue($L, q, \mu(q)$) ; mark $q$
```

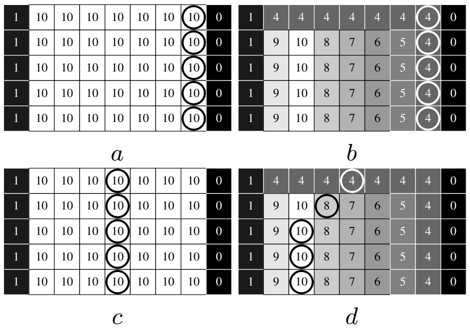


Fig. 4. a, b: two images with a possible W-crest (circled) computed thanks to procedure **M-watershed** with a constant priority function. c, d: the same two images with the W-crest (circled) computed thanks to procedure **M-watershed** with the lexicographic priority function

The following property is a consequence of results proved in [5] and of the fact that, obviously, each point is selected at most once by this algorithm.

Property 2. *Whatever the chosen priority function, the output of Procedure* **M-watershed** *is an M-watershed of the input.*
Let n and m denote respectively the number of vertices and the number of arcs in the graph (E, Γ). The time complexity of Procedure **M-watershed** *is in $O(n + m) + k$, where k is the overall complexity for the management of the priority queue.*

We introduced the priority function and the priority queue in order to take into account some geometrical criteria. For example, with a constant priority function, plateaux or even domes located between basins may be thinned in different ways, depending on the arbitrary choices that are allowed by the calls to **ExtractPrioQueue** with this particular priority function (line 05). See Fig. 4 a, b for a possible result of procedure **M-watershed** with a constant priority function.

In order to "guide" the watershed set towards the highest locations of the domes and the "center" of the plateaux, we choose a lexicographic priority function μ described below.

Let $F \in \mathcal{F}(E)$, let d be a distance on E (*e.g.*, the Euclidean distance), let $p \in E$. We denote by $D(p)$ be the minimal distance between p and any point q strictly lower than p, that is, $D(p) = \min\{d(p, q); F(q) < F(p)\}$.

It is easy to build a function μ such that, for any p, q in E:

- if $F(p) < F(q)$ then $\mu(p) > \mu(q)$;
- if $F(p) = F(q)$ and $D(p) \leq D(q)$ then $\mu(p) \geq \mu(q)$.

See Fig. 4 c, d for the result of **M-watershed** with such a priority function.

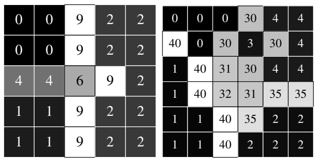


Fig. 5. Examples of W-destructible points in an MS-watershed which are neither $\tilde{\text{M}}$-points nor $\tilde{\text{S}}$-points: the point at 6 in the image on the left, the points at 31 and 32 in the image on the right

The values of such a priority function may be pre-computed by a linear-time algorithm, see for example algorithm 4.5 of [9] which is called *lower completion* algorithm. The efficient management of priority queues is a well studied problem, and efficient solutions exist (see e.g. [11]). Furthermore, in most current situations of image analysis, where the number of possible values for the priority function is limited and the number of neighbors of a point is a small constant, specific linear algorithms can be used, avoiding the use of a priority queue. An example of such a linear algorithm is given in the next section, with algorithm **TopologicalWatershed**.

4 Watershed Algorithm

After iteratively lowering $\tilde{\text{M}}$-points until stability, we have to process the other W-destructible points in order to get a topological watershed. Let $F \in \mathcal{F}(E)$, let us call an *MS-watershed of* F a map obtained from F by iteratively lowering $\tilde{\text{M}}$-points and $\tilde{\text{S}}$-points until stability. We could think that all $\tilde{\text{P}}$-points will be eventually changed to $\tilde{\text{M}}$-points and then M-lowered in such a process, as it is the case for images like Fig. 3*a*. But the examples of Fig. 5 show that it is not always the case, in other words, an MS-watershed of F is not always a topological watershed of F. Furthermore, there may exist thick regions made of $\tilde{\text{P}}$-points in an MS-watershed, and although $\tilde{\text{M}}$-points and $\tilde{\text{S}}$-points may be lowered directly down to their lowest possible value, we have no such guarantee for the $\tilde{\text{P}}$-points (see theorem 5 of [5]).

Thus, we must propose a criterion for the selection of the remaining W-destructible points, in order to avoid multiple selections of the same point. The idea is to give the greatest priority to a W-destructible point which may be lowered down to the lowest possible value. We prove that an algorithm which uses this strategy never selects the same point twice. A priority queue could be used, as in the previous section, to select W-destructible points in the appropriate order. Here, we propose a specific linear watershed algorithm which may be used when the grayscale range is small. Let $F \in \mathcal{F}(E)$, let $k_{\min} = \min\{F(p); p \in E\}$ and $k_{\max} = \max\{F(p); p \in E\}$.

```
Procedure TopologicalWatershed (Input $F$ ; Output $F$)
01.     For $k$ From $k_{\min}$ To $k_{\max}$ Do $L_k \leftarrow \emptyset$
02.     For All $p \in E$ Do
03.         $i \leftarrow$ W-Destructible($F$, $p$)
04.         If $i \neq \infty$ Then
05.             $L_i \leftarrow L_i \cup \{p\}$ ; $K(p) \leftarrow i$
06.     For $k = k_{\min}$ To $k_{\max}$ Do
07.         While $\exists p \in L_k$ Do
08.             $L_k = L_k \setminus \{p\}$
09.             If $K(p) = k$ Then
10.                 $F(p) \leftarrow k$
11.                 For All $q \in \Gamma(p)$, $k < F(q)$ Do
12.                     $i \leftarrow$ W-Destructible($F$, $q$)
13.                     If $i = \infty$ Then $K(q) \leftarrow \infty$
14.                     Else If $K(q) \neq i$ Then
15.                         $L_i \leftarrow L_i \cup \{q\}$ ; $K(q) \leftarrow i$
```

We have the following guarantees:

Property 3. *In algorithm* **TopologicalWatershed***,*
i) at the end of the execution, F is a topological watershed of the input map;
ii) let n and m denote respectively the number of vertices and the number of arcs in the graph (E, Γ). If $k_{max} - k_{min} \leq n$, then the time complexity of the algorithm is in $O(n + m)$.

As discussed in the previous section, this algorithm provides topological guarantees but does not care about geometrical criteria. If we want to take such criteria into account, we can use first the procedure **M-watershed** with the priority function described at the end of section 3, and then the procedure **TopologicalWatershed**.

5 Watershed from Markers

In many applications, instead of finding a separation between the minima of the input function, we need to separate the components of a given set of points called the *marker*. Let us illustrate how to reach this goal using the topological watershed, following a classical approach based on reconstruction.

Fig. 6 illustrates the outline of the whole procedure. Fig. 6*a* shows the input data, function F and marker M. Fig. 6*b*: a function G is generated, such that $G(x) = k_{\min}$ for all $x \in M$, and $G(x) = k_{\max}$ for all $x \notin M$. The function $F' = \min(F, G)$ is computed. Fig. 6*c*: we compute the morphological geodesic reconstruction G' of G over F'. See [13] for a description of this operator, which can be efficiently implemented thanks to the component tree. Notice that, by construction, each minimum of G' contains a component of M. Fig. 6*d*: finally, an M-watershed W of G' is extracted, hence, a W-crest C.

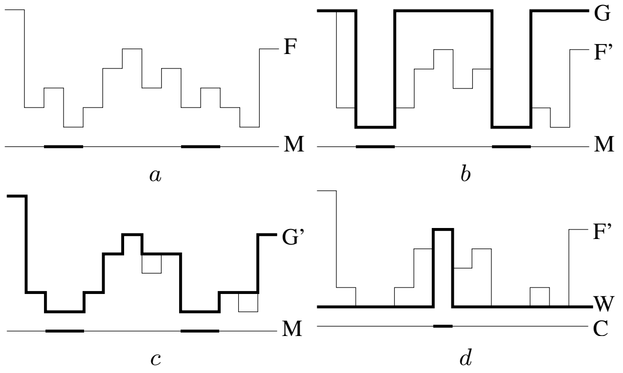


Fig. 6. Outline of a topological watershed-from-markers procedure

References

1. G. Bertrand, "On topological watersheds", *Journal of Mathematical Imaging and Vision*, Vol. 22, pp. 217-230, 2005.
2. S. Beucher, Ch. Lantuéjoul, "Use of watersheds in contour detection", *Proc. Int. Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation*, Rennes, France, 1979.
3. S. Beucher, F. Meyer, "The morphological approach to segmentation: the watershed transformation", *Mathematical Morphology in Image Processing*, Chap. 12, pp. 433-481, Dougherty Ed., Marcel Dekker, 1993.
4. M. Couprie, G. Bertrand, "Topological grayscale watershed transformation", *Proc. SPIE Vision Geometry VI*, Vol. 3168, pp. 136-146, 1997.
5. M. Couprie, L. Najman, G. Bertrand, "Quasi-linear algorithms for the topological watershed", *Journal of Mathematical Imaging and Vision*, Vol. 22, pp. 231-249, 2005.
6. F. Meyer, "Un algorithme optimal de ligne de partage des eaux", *Proc. 8th Conf. Reconnaissance des Formes et Intelligence Artificielle*, Vol. 2, pp. 847-859, AFCET Ed., Lyon, 1991.
7. L. Najman, M. Couprie, "Quasi-linear algorithm for the component tree", *Proc. SPIE Vision Geometry XII*, Vol. 5300, pp. 98-107, 2004.
8. L. Najman, M. Couprie, G. Bertrand, "Watersheds, extension maps, and the emergence paradigm", report IGM2004-04 of the Institut Gaspard Monge (University of Marne-la-Vallée), to appear in *Discrete Applied Mathematics*, 2004.
9. J. Roerdink, A. Meijster, "The watershed transform: definitions, algorithms and parallelization strategies", *Fundamenta Informaticae*, Vol. 41, pp. 187-228, 2000.
10. R.E. Tarjan, "Disjoint sets" *Data Structures and Network Algorithms*, Chap. 2, pp. 23-31, SIAM, 1978.
11. M. Thorup, "On RAM priority queues", *7th ACM-SIAM Symposium on Discrete Algorithms*, pp. 59-67, 1996.
12. L. Vincent, P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations", *IEEE Trans. on PAMI*, Vol. 13, No. 6, pp. 583-598, 1991.
13. L. Vincent, "Morphological Grayscale Reconstruction in Image Analysis: Application and Efficient Algorithms", *IEEE Trans. on PAMI*, Vol. 2, No. 2, pp. 176-201, 1993.