# A watershed algorithm progressively unveiling its optimality

Fernand Meyer

# A watershed algorithm progressively unveiling its optimality

Fernand Meyer

CMM-Centre de Morphologie Mathématique,
Mathématiques et Systèmes, MINES ParisTech, PSL Reseach University, Paris,
France fernand.meyer@mines-paristech.fr

**Abstract.** In 1991 I described a particularly simple and elegant watershed algorithm, where the flooding a topographic surface was scheduled by a hierarchical queue. In 2004 the watershed line has been described as the skeleton by zone of influence for the topographic distance. The same algorithm still applies. In 2012 I defined a new distance based on a lexicographic ordering of the downstream paths leading each node to a regional minimum. Without changing a iota, the same algorithm does the job..

KEYWORDS: Watershed, flooding, hierarchical queue, topographic distance, weighted graphs, steepest lexicographic distance

## 1 Introduction

This paper is singular as it analyses a paper published in 1991 [5], known and used by many, describing an algorithm for constructing watershed partitions (for a literature review on the watershed, see [9]). The title of this paper was rather presomptuous "un algorithme optimal de ligne de partage des eaux", which means "an optimal watershed algorithm". In the sequel we refer to it as the "HQ algo."

Classically considering a grey tone function as a topographic surface, the algorithm constructs a watershed partition by flooding this surface, the darker levels being flooded before the brighter levels. At the same time it provides a correct treatment of the plateaus of uniform grey tone: the flooding starts at its downward boundary and progresses inwards with uniform speed. The flooding is controlled by a hierarchical queue, i.e. a series of hierarchical FIFO queues. Each queue is devoted to the flooding of a particular level. The FIFO structure cares for the flooding of the plateaus.

If one considers by which other pixel each pixel is flooded, one obtains a flooding trajectory. A pixel is assigned to the catchment basin of a particular minimum $m$ if there exists a flooding trajectory between this pixel and $m$. A pixel may be linked to several minima by distinct flooding trajectories. The hierarchical queues selects a particular flooding trajectory for each pixel and assigns this pixel to the minimum at the beginning of the trajectory. The quality of the final watershed partition entirely relies on the choice of the best trajectories.

The algorithm has been developed for a lexicographic order of flooding: between two pixels, the first to be flooded is the lowest ; and if they have the same gray tone, the first to be flooded is the nearest to the lower boundary of a plateau.

Later the topographic distance has been developed and a more selective order relation defined. If a node has lower neighbors, it may be flooded only by one of the lowest of them. The behaviour on plateaus remains identical. The "HQ algo". remains valid for this new lexicographic distance derived from the topographic distance [8,6].

Finally, in 2012, I introduced a new distance for which the geodesics are the steepest possible [7]. The steepest paths are chosen among the smallest for a lexicographic order relation based on the gray tone values of all pixels along the path. It was a real surprise to discover that, again, without the slightest modification, the algorithm of 1991 does the job.

The paper explains why the HQ algorithm remains valid for these 3 types of very different distances. A slight modification of the HQ algorithm permits to prune the graph in order to keep only the infinitely steep flooding paths. We conclude by some applications using these paths.

We conclude, recognizing that the HQ algorithm was, indeed, in more senses than suspected, optimal.

## 2 The early days of the watershed

### 2.1 The geodesic distance and the skeleton by zones of influence

Christian Lantuejoul, in order to model a polycrystalline alloy, defined and studied the skeleton by zones of influence of a binary collection of grains in his thesis [3] ; he studied the geodesic metric used for constructing a SKIZ in [4]. The shortest distance $d_Z(x, y)$ between two pixels $x$ and $y$ in a domain $Z$ is the length of the shortest path linking both pixels within the domain. If $X$ is a subset of $Z$, we likewise define the geodesic distance between a point $x \in Z$ and the set $X$, as the shortest path between $x$ and a pixel belonging to $X$.

Consider now a set $X \subset Z$, union of a family of connected components $X_i$. The skeleton by zone of influence of $X$, or skiz$(X, Z)$ assigns to each connected component $X_i$ the pixels which are closer to $X_i$ than to any other set $X_j$ for the geodesic distance $d_Z$.

The first implementation of the SKIZ was made on the TAS, a hardwired processor developed at the CMM and used homotopic thickenings for constructing the SKIZ. With the advent of cheaper memories and the personal computer, it was possible to represent images on random access memories (whereas the TAS permitted only raster scan access to the images). Luc Vincent proposed a very efficient implementation of a SKIZ [10]; the growing of the various germs being governed by a FIFO. The algorithm for constructing the SKIZ skiz$(X, Z)$ of a family of particles $X = (X_i)$ within a domain $Z$ is the algorithm 1.

**Algorithme 1:** A fifo based algorithm for constructing the geodesic SKIZ

**Input** : A family of seeds $(K_i)_{i \in I}$ within a domain $Z$

**Result** : The geodesic SKIZ of the of seeds $(K_i)_{i \in I}$ within $Z$

**Initialisation**: Create a FIFO $Q$
Introduce the inside boundary pixels of the seeds $K_i$ in $Q$

**while** *$Q$ not empty* **do**
  extract the node $j$ with the highest priority from $Q$
  **for** *each unlabeled neighboring node $i \in Z$ of $j$* **do**
    $label(i) = label(j)$
    put $i$ in the queue $Q$

## 2.2 The level by level construction of the watershed

The watershed partition $\pi_f$ of a grey tone image represented by a function $f$, defined on a domain $D$ and taking its value in the interval $[0, N]$ is then easily obtained with the algorithm 2, by iteratively constructing a series of SKIZ in increasing domains depending on the successive thresholds of $f$ [1]. In algorithm 2, the notation $\{f = \lambda\}$ means the set of pixels of $D$ for which $f$ takes the value $\lambda$

**Algorithme 2:** A level by level watershed algorithm

**Input** : A function $f$

**Result** : the watershed partition $\pi_f$ of $f$

**Initialisation**: Initialize the set $\pi_f$ with the regional minima of $f$

**for** *each gray level $\lambda = 1, N$* **do**
  $\pi_f = skiz(\pi_f, \pi_f \cup \{f = \lambda\})$

One gets the watershed algorithm by introducing the algorithm for the SKIZ into the algorithm of C. Lantuéjoul. L. Vincent and P.Soille completed the algorithm by adding a clever mechanism for finding the successive thresholds of a grey tone function [10].

## 2.3 A hierarchical queue algorithm for the watershed

The algorithm 2 repeats the SKIZ construction for all successive levels of the grey tone function sequentially. For each level it uses the algorithm 1. When a pixel is dequeued from the FIFO, its neighbors are labeled if they belong to the same level set, and discarded if not. They will be processed later, when the level set corresponding to their greytone value is processed. The principal innovation of the "HQ algo" [5] is to create all FIFOs at once, each being devoted to a distinct grey tone, with the advantage to to put all pixels which are met in a waiting position before they flood neighboring pixels. When a node $x$ is dequeued all its neighbors may be processed: the neighbors with the same grey tone as $x$ are put in the same FIFO as $x$, the others are put in the FIFOs corresponding to their grey tone. The resulting algorithm 3 is particularly simple.

**Algorithme 3:** The hierarchical queue based watershed algorithm

**Input** : A gray tone function

**Result** : A partition of labeled catchment basins

**Initialisation**: Detect and label the regional minima
Create a hierarchical queue HQ
Introduce the inside boundary pixels of the minima in the HQ

**while** *HQ not empty* **do**
- extract the node $j$ with the highest priority from the HQ
- **for** *each unlabeled neighboring node $i$ of $j$* **do**
  - $label(i) = label(j)$
  - put $i$ in the queue with priority $f_i$

**Analysis of the algorithm** The nodes are processed according to a dual order: nodes with a lower altitude are flooded before nodes with a higher altitude. And nodes within a plateau are processed in an order proportional to the distance to the lower border of the plateau.

# 3 The watershed as the SKIZ for the topographic distance

Each image defined on a grid may be considered as a particular graph. The pixels are the nodes of the graph ; neighboring pixels are linked by a node. All algorithms defined so far immediately to node weighted graphs. We now give a few reminders on graphs

## 3.1 Reminders on node weighted graphs

A *non oriented graph* $G = [N, E]$ contains a set $N$ of vertices or nodes and a set $E$ of edges ; an edge being a pair of vertices. The nodes are designated with small letters: $p, q, r$...The edge linking the nodes $p$ and $q$ is designated by $e_{pq}$.

*A path*, $\pi$, is a sequence of vertices and edges, interweaved in the following way: $\pi$ starts with a vertex, say $p$, followed by an edge $e_{pq}$, incident to $p$, followed by the other endpoint $q$ of $e_{pq}$, and so on.

Denote by $\mathcal{F}_n$ the sets of non negative weight functions on the nodes. The function $\nu \in \mathcal{F}_n$ takes the weight $\nu_p$ on the node $p$.

A subgraph $G'$ of a node weighted graph $G$ is a *flat zone*, if any two nodes of $G'$ are connected by a path along which all nodes have the same weight. If furthermore all neighboring nodes have a higher altitude, it is a *regional minimum*. A *flooding path* is a path along which the node weights is never increasing.

**Definition 1.** *The catchment zone of a minimum $m$ is the set of nodes linked by a flooding path with a minimum.*

Obviously, a node may be linked with several regional minima through distinct flooding paths. The catchment zones may overlap and do not necessarily form a partition. They form a partition if each node is linked with one and only one basin through a flooding path. As there may be several paths towards distinct minima, the "HQ algo." selects for each node a particular flooding path linking this node with are regional minimum. The next section shows how to restrict even more the admissible flooding paths.

### 3.2 The topographic distance

Consider an arbitrary path $\pi = (x_1, x_2, ..., x_p)$ of the node weighted graph $G$ between two nodes $x_1$ and $x_n$. The weight $\nu_p$ at node $x_p$ can be written:

$\nu_p = \nu_p - \nu_{p-1} + \nu_{p-1} - \nu_{p-2} + \nu_{p-2} - \nu_{p-3} + .... + \nu_2 - \nu_1 + \nu_1$

The node $k-1$ is not necessarily the lowest node of node $k$, therefore $\nu_{k-1} \geq \varepsilon_n \nu_k$ and $\nu_k - \nu_{k-1} \leq \nu_k - \varepsilon_n \nu_k$.

Replacing each increment $\nu_k - \nu_{k-1}$ by $\nu_k - \varepsilon_n \nu_k$ will produce a sum $\nu_p - \varepsilon_n \nu_p + \nu_{p-1} - \varepsilon_n \nu_{p-1} + .... + \nu_2 - \varepsilon_n \nu_2 + \nu_1$ which is larger than $\nu_p$. It is called the topographic length of the path $\pi = (x_1, x_2, ..., x_p)$.

The path with the shortest topographic length between two nodes is called the topographic distance between these nodes. It will be equal to $\nu_p$ if and only if the path $(x_1, x_2, ..., x_p)$ precisely is a path of steepest descent, from each node to its lowest neighbor. The topographic distance has been introduced independantly in [8,6].

Consider again the "HQ algo". The regional minima are labeled. During the execution of the algorithm, each node $p$, when dequeued, assigns its label to its neighbors without label. If a node $q$ has no label when its neighboring node $p$ is dequeued, it means that $q$ has no other neighbor which has been dequeued before. If the nodes $p$ and $q$ have distinct weights, it means that $p$ is one of its lowest neighbors: between nodes with distinct grey tones, it is the topographic distance which prevails. If the nodes $p$ and $q$ have the same weight, it means that $p$ and $q$ both belong to the same plateau and $p$ is closer to the lower border of the plateau than $q$. We obtain like that a more restrictive lexicographic distance, where the first term is the topographic distance and the second term the distance to the lower boundaries of the plateaus.

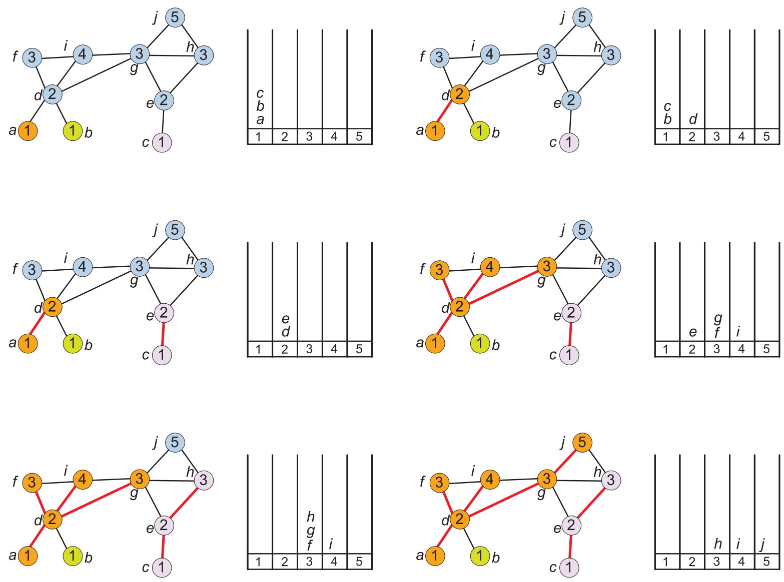


**Fig. 1.** The hierarchical queue algorithm constructing the watershed partition of a node weighted graph

### 3.3 Illustration of the HQ algorithm applied to node weighted graphs.

Fig.1 shows how the hierarchical queue algorithm constructs the watershed partition of a node weighted graphs.

A hierarchical queue is created. The regional minima, the nodes $a, b$ and $c$ are detected and labeled each with a distinct colour ; having the grey tone 1, they are put into the FIFO of priority 1.

The first node to be extracted is $a$. Its neighboring node $d$, has no label ; $d$ gets the label of $a$ and is introduced in the FIFO of priority 2.

The node $b$ is extracted. It has no neighbor. The node $c$ propagates its label to $e$, which is introduced in the FIFO of priority 2.

The node $d$ gives its labels to its unlabeled neighbors $f$, $g$ and $i$. The nodes $f$ and $g$ are introduced in the FIFO of priority 3 and the node $i$ in the FIFO of priority 4.

The node $f$ has no unlabeled neighbor. The node $g$ gives its label to the node $j$ ; the node $j$ is introduced in the FIFO with a priority 5 as shown in the last configuration of the HQ in fig.1. All nodes of the graph are now labeled and constitute the watershed partition of the graph. The last nodes are extracted from the HQ without introducing new nodes in the HQ. When the HQ is empty, the algorithm stops.

## 4 The watershed as the SKIZ for infinitely steep lexicographic distance

The hierarchical queue algorithm presented so far makes 2 jobs. The first job is selecting a family of flooding paths which are all geodesics of the same distance function. If several geodesics link a node to several regional minima, it furthermore selects one of them, the first which is able to label a node without label.

We now establish, that in fact the HQ algorithm makes a much more severe selection between flooding paths as it appears so far. It does in fact implement a much more selective distance function, based on an infinite lexicographic order.

### 4.1 A lexicographic preorder relation between steepest path

Consider now a flooding path, i.e. a path along which the node weights are never increasing, ending at a node $m$ belonging to a regional minimum. If $m$ is an isolated regional minimum node, we artificially add a loop edge linking the node $m$ with itself. Like that the flooding path, as it reaches the node $m$ may be prolongated into a path of infinite length ; if $m$ is an isolated regional minimum node, the cycles around in the added loop edge ; if $m$ has a neighboring node $p$ belonging to the same regional minimum, the path oscillates between $m$ and $p$.

A **lexicographic preorder relation** compares the infinite paths $\pi = (p_1, p_2, ...p_k, ...)$ and $\chi = (q_1, q_2, ...q_k, ...)$:

* $\pi \prec \chi$ if $\nu_{p_1} < \nu_{q_1}$ or there exists $t$ such that $\begin{array}{l} \forall l < t : \nu_{p_l} = \nu_{q_l} \\ \nu_{p_t} < \nu_{q_t} \end{array}$

* $\pi \preceq \chi$ if $\pi \prec \chi$ or if $\forall l : \nu_{p_l} = \nu_{q_l}$.

The preorder relation $\preceq$ is total, as it permits to compare all paths with the same origin $p$. The lowest of them are called the *steepest* paths of origin $p$. It is obvious that if $\pi$ is a steepest path of origin $p$ towards a regional minimum $m$, then any subpass obtained by suppressing a given number of nodes at the beginning of the path also is a steepest path.

A node may be linked by flooding paths with 2 or more distinct regional minima. In this case the watershed zones overlap. If one considers only the *steepest* paths, this will rarely happen, as the weights all along the paths should be absolutely identical. In particular, steepest paths reaching regional minima with different altitude necessarily have a distinct steepness. One obtains like that highly accurate watershed partitions, whereas the classical algorithms, being myopic as they use only the adjacent edges of each node, pick one solution out of many.

### 4.2 The infinitely steep lexicographic distance

We define the infinitely steep lexicographic distances (in short $\infty sld$) between nodes and regional minima as follows. The distance $\chi(p, m)$ between a node $p$ and a regional minimum $m$ is equal to $\infty$ if there exists no steepest path between $p$ and a node belonging to $m$. Otherwise it is equal to the sequence of never increasing weights of the nodes along a steepest path linking $p$ and a node belonging to $m$.

The infinitely steep skeleton by influence is the SKIZ associated to this distance [7]. Surprisingly enough, the HQ algo.precisely implements the infinitely steep lexicographic distance.

We write $\chi_p$ for the $\infty sld$ of each node. It is an infinite series of weights, the first in the series being the weight of the node $p$ itself.

For a regional minimum node $m$, this weight is equal to an infinite series of identical values $\nu_m$. As the regional minima nodes are introduced in the FIFO with the same priority as their weights, they are also ranked according to their distances $\chi_m$.

Suppose now that when a node $p$ is extracted from the HQ, all nodes labeled so far have got their correct $\infty sld$ distance. The $\infty sld$ distance of $p$ is $\chi_p$ and has been correctly estimated. If $q$ is a neighboring node of $q$ without label, this means that $p$ is the neighbor of $q$ with the lowest $\infty sld$ distance. The $\infty sld$ distance of $q$ is then obtained by appending the weight $\nu_q$ of $q$ at the beginning of the $\infty sld$ distance of $p$ : $\chi_p = \nu_q \rhd \chi_p$. The node $q$ gets the same label as the node $p$, and is introduced in the FIFO with the priority equal to its weight $\nu_q$. It takes a place in this FIFO after all nodes with the same weight but with lower $\infty sld$ distances and before all nodes with the same weight but with higher $\infty sld$ distances.

The order in which each node floods its neighbors is much more subtle as imagined in 1991: the nodes flood their neighbors according to their $\infty sld$ distances. The order in which the plateaus are flooded is only a particular case. When a plateau is flooded, its lower boundary pixels are not equivalent, but enter into play according to their own $\infty sld$ distances.

### 4.3 The watershed for edge weighted graphs

There are situations where one desires obtaining the watershed partition associated to an edge weighted graph. For instance, the relations between the catchment basins of a topographic surface also may be modelled as a graph. The catchment basins are the nodes of the graph : neighboring basins are linked by an edge, which is weighted by the altitude of the pass point leading from one basin to the other.

The definition of regional minima, flooding paths and catchment basins are similar to those defined for node weighted graphs.

Denote by $\mathcal{F}_e$ the sets of non negative weight functions on the edges. The function $\eta \in \mathcal{F}_e$ takes the value $\eta_{pq}$ on the edge $e_{pq}$, and the graph holding only edge weights is designated by $G(\eta, nil)$.

A subgraph $G'$ of an edge weighted graph $G$ is a *flat zone*, if any two nodes of $G'$ are connected by a path with uniform edge weights. If in addition, all edges in its cocycle have a higher altitude, it is a *regional minimum.* A path $\pi = \{v_1.e_{12}.v_2.e_{23}.v_3 \cdots\}$ is a *flooding path*, if each edge $e_k = (v_k, v_{k+1})$ is **one**

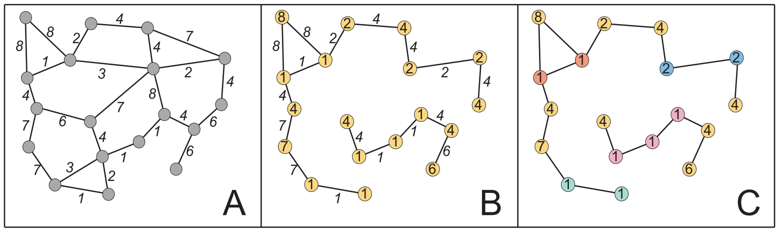


**Fig. 2.** A: an edge weighted graph
B: each edge which is not the lowest edge of one of its extremities has been suppressed. Each node is assigned a weight equal to the weight of its lowest adjacent edge.
C: The edge weights have been dropped and the regional minima detected and labeled.

**of the lowest edges of its extremity** $v_k$, and if along the path the edge weights are never increasing.

**Definition 2.** *The catchment zone of a regional minimum* $m$ *is the set of nodes linked by a flooding path having its end-node within* $m$.

Jean Cousty et al. studied the watershed on edge weighted graphs, calling it watershed cuts, as each catchment basins is represented by a connected subgraph or a tree [2]. Cousty found that in some respects, the watershed on edge weighted graphs is superior that the watershed on node weighted graphs.

In fact, there is no such superiority and I established in [7] the perfect equivalence of node or edge weighted graphs with respect to the watershed. Figure 2 shows how to transform an edge weighted graph into a node weighted graph having the same regional minima, flooding paths and catchment zones as the initial edge weighted graph. Figure 2A presents an edge weighted graph. The edges which are not the lowest of one of their extremities do not belong to any flooding path. In figure 2B, each edge which is not the lowest edge of one of its extremities has been suppressed. Each node is assigned a weight equal to the weight of its lowest adjacent edge. Finally in figure 2C, the edge weights have been dropped and the regional minima detected and labeled. The HQ algorithm for node weighted graphs may be applied on this graph, yielding the correct watershed partition for the initial edge weighted graph.

### 4.4 Constructing the steepest subgraph of a node or edge weighted graph

We have established that the HQ algorithm choses among al flooding paths the steepest. If there are 2 paths with identical weights leading to 2 distinct minima, it does an arbitrary choice between both for constructing a watershed partition.

It is possible to slightly modify the HQ algorithm for constructing a subgraph $\widetilde{G}$ with the same nodes as the initial graph but without any edge which does not belong to a steepest path. We have seen that all nodes with the same

**Algorithme 4:** Construction of the steepest partial subgraph of $G$

**Input** : $G = [N, E]$, a node weighted graph

**Result** : $\widetilde{G} = [N, \widetilde{E}]$, the steepest partial graph of $G$

**Initialisation**: $HQ$ = hierarchical queue
Create a set $S$ containing the regional minima of $G$
Put the inside boundaries of the regional minima in $HQ$
Create $\widetilde{G} = [N, \varnothing]$
Close the FIFOs

**while** *$HQ$ not empty* **do**
  extract all nodes up to the first tag into a set $P$
  Suppress this tag
  **for** *each $q \in \partial^+ P \cap \overline{S}$* **do**
    **for** *each $p \in P$, such that $(p, q)$ neighbors* **do**
      $\widetilde{E} = \widetilde{E} \cup e_{qp}$
    $S = S \cup \{q\}$
    put $q$ in the queue with a priority equal to its weight
  Close all FIFOs

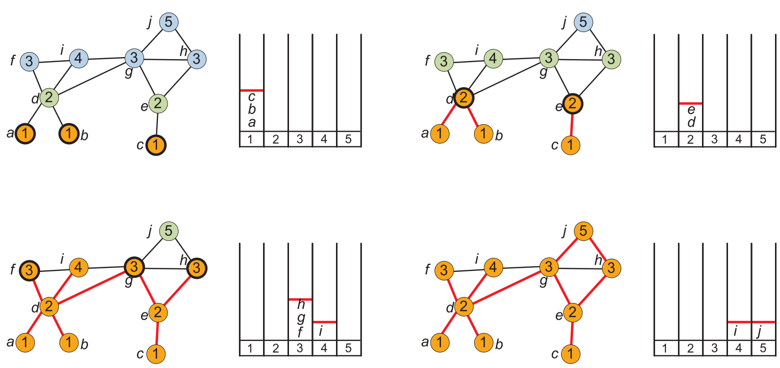

**Fig. 3.** A hierarchical queue algorithm for constructing the steepest paths (bold red edges) in a node weighted graph.

lexicographic distance to a minimum are regrouped in the same FIFO before they flood their unflooded neighbors. We will introduce tags in the FIFOs of the HQ in order to separate between 2 tags all nodes with the same $\infty sld$ distance. In algorithm 4, the expression "close the FIFOs" means introduce a tag in all non empty FIFOs which are without a tag on top.

At initialization a HQ is created. The boundary nodes of the regional minima are introduced in a set $S$ and in the FIFOs corresponding to their weight and all non empty FIFOs are closed. The set $S$ will contain the union of all nodes which are or have been in the HQ. The algorithm stops when $S$ contains all nodes of $N$.

As long as the hierarchical queue is not empty, all nodes with the same highest prioriry are extracted, including the tag itself. They are put into a set $P$. We call $\partial^+ P$ the set of nodes which are not in $P$ but have a neighbor in $P$. The set $\partial^+ P \cap \overline{S}$ represents the nodes of $\partial^+ P$ which are not yet in the set $S$. For each couple of nodes $(p, q)$, $p \in \partial^+ P \cap \overline{S}$ and $q \notin S$, the edge $e_{pq}$ is added to the edges $\widetilde{E}$ of the graph $\widetilde{G}$ (these edges are indicated in red in fig.3) . The complete algorithm is described in algorithm 4.

Ilustration of the algorithm (fig.3):

A hierarchical queue is created. The boundary nodes of the regional minima, here the nodes $a$, $b$ and $c$ are put in the HQ. A tag (red line) is added to the FIFO.

The first set of most priority nodes $(a, b, c)$ are extracted from the HQ including the tag in top of them. The node $d$ is neighbor of $a$ and of $b$. The edges $e_{ad}$ and $e_{bd}$ are created in $\widetilde{E}$. The node $e$ is neighbor of the node $c$. An edge $e_{ce}$ is created in $\widetilde{E}$. The nodes $d$ and $e$ are put in the FIFO of priority 2 and in the set $S$; this FIFO is then closed, by adding a tag (second figure in fig.3). The next group of nodes below a tag to be extracted are the nodes $d$ and $e$. The node $d$ has three neighbors $(f, i, g)$. The node $g$ is also neighbor of the node $e$. The node $h$ is a second neighbor of $e$. The edges $e_{df}, e_{di}, e_{dg}$ and $e_{eg}, e_{eh}$ are created in $\widetilde{E}$, the nodes $f, i, g, h$ are introduced in the HQ in their corresponding FIFOs and in the set $S$. The open FIFOs are then closed (third figure in fig.3). The next group of nodes are $(f, g, h)$. The nodes $g$ and $h$ have a neighbor outside $S$, the node $j$ The edges $e_{gj}, e_{hj}$ are created in $\widetilde{E}$, and the node $j$ introduced in the $HQ$ and in $S$. The node $i$ has no neighbor outside $S$. At this point all nodes of $N$ are inside $S$. The algorithm will stop when the last nodes are extracted without sending new nodes in the HQ.

*Application:* Fig.4A represents a digital elevation model. In Fig.4B a number of points have been hand marked. Following the $\infty - steepest$ paths starting from these points creates the upstream of the rivers. It is important for this application that all steepest paths are detected, in order to avoid any bias.

## 5 Conclusion

The HQ algorithm has been developed with a given definition of the watershed in mind and revealed to be still valid for two different mathematical formulations

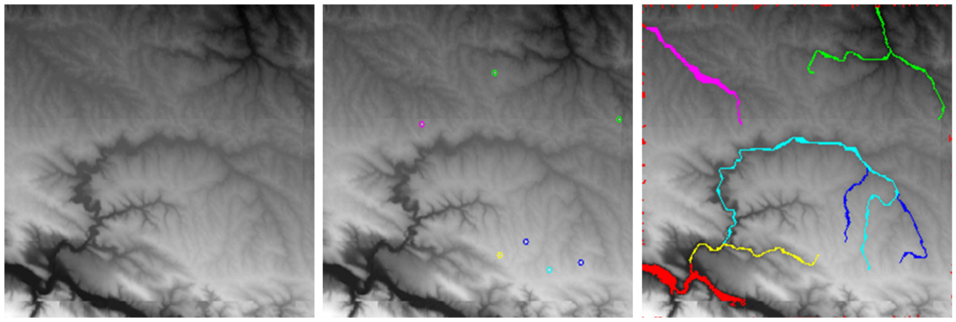

**Fig. 4.** Left : a digital elevation model
Center: Hand marked extremities of rivers
Right: The union of the steepest paths having these nodes as extremities.

of the watershed. This paper illustrates the major and often ignored role of the data structure used for implementing a mathematical concept: among all possible solutions compatible with a particular definition, it does a hidden sampling.

## References


1. S. Beucher and C. Lantuéjoul. Use of watersheds in contour detection. In *Watersheds of Functions and Picture Segmentation*, pages 1928–1931, Paris, May 1982.
2. Jean Cousty, Gilles Bertrand, Laurent Najman, and Michel Couprie. Watershed cuts: Minimum spanning forests and the drop of water principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:1362–1374, 2009.
3. C. Lantuéjoul. *La squelettisation et son application aux mesures topologiques de mosaßques polycristallines.* PhD thesis, École nationale supérieure des mines de Paris, 1978.
4. C. Lantuéjoul and S. Beucher. On the use of the geodesic metric in image analysis. *J. Microsc.*, 1981.
5. F. Meyer. Un algorithme optimal de ligne de partage des eaux. In *Proceedings* $8^{\underline{ème}}$ *Congrès AFCET, Lyon-Villeurbanne*, pages 847–857, 1991.
6. F. Meyer. Topographic distance and watershed lines. *Signal Processing*, pages 113–125, 1994.
7. Fernand Meyer. Watersheds on weighted graphs. *Pattern Recognition Letters*, 2014.
8. Laurent Najman and Michel Schmitt. Watershed of a continuous function. *Signal Processing*, 38(1):99 – 112, 1994. Mathematical Morphology and its Applications to Signal Processing.
9. L. Najman and H. Talbot (ed) Mathematical morphology *Wiley editor*, 2012
10. Soille Pierre Vincent, Luc. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.