The Waterfall Hierarchy on Weighted Graphs

Fernand Meyer(✉)

CMM – Centre de Morphologie Mathématique,
MINES ParisTech – PSL Research University,
35 rue St Honoré, 77300 Fontainebleau, France
fernand.meyer@mines-paristech.fr

Abstract. We study and present two new algorithms for constructing the waterfall hierarchy of a topographic surface. The first models a topographic surface as a flooding graph, each node representing a lake filling a catchment basin up to its lowest pass point ; each edge representing such a pass point. The second algorithm produces the waterfall partition in one pass through the edges of a minimum spanning tree of the region adjacency graph associated to a topographic surface.

Keywords: Waterfall · Watershed · Edge and node weighted graphs · 1 Pass waterfall algorithm

1 Introduction

The waterfall hierarchy represents the nested structure of the catchment basins of a topographic surface [3]. Consider a series of increasing floodings of a topographic surface. The associated watershed partitions form a hierarchy, as each catchment basin of a higher flooding is the union of catchment basins of a lower flooding of the surface. The waterfall hierarchy is obtained when a topographic surface is submitted to the highest flooding possible without any overflow from one catchment basin to a neighboring bassin ; each basin is then filled up to its lowest pass leading to a neighboring basin. This flooding level is thus independent of the shape, size or depth of the basins. As a result is the waterfall hierarchy invariant under the change of scale, orientation or contrast of the images.

Fig.1 presents a topographic surface with 3 successive levels of waterfall flooding. The associated waterfall partitions are indicated below each flooded surface.

The waterfall hierarchy unfolds the nested structure of the catchment basins of a topographic surface. From one level to the next of the waterfall hierarchy, a number of catchment basins merge, revealing a new scale, until a last flooding completely covers the domain. In general does the waterfall hierarchy possess only a low number of hierarchical levels. The lowest levels provide highly simplified partitions compared to the initial oversegmentation produced by the watershed. The waterfall hierarchy has been used with success for segmenting not only gray tone images [11], but also color and texture images [19]. It has been used not only on images but also on probability distribution functions used for the stochastic watershed of multipectral images [2] or on triangular meshes [9]. The domain of application is broad going from multimedia images [22] to SAR imagery [20].

J.A. Benediktsson et al. (Eds.): ISMM 2015, LNCS 9082, pp. 325–336, 2015.
DOI: 10.1007/978-3-319-18720-4_28

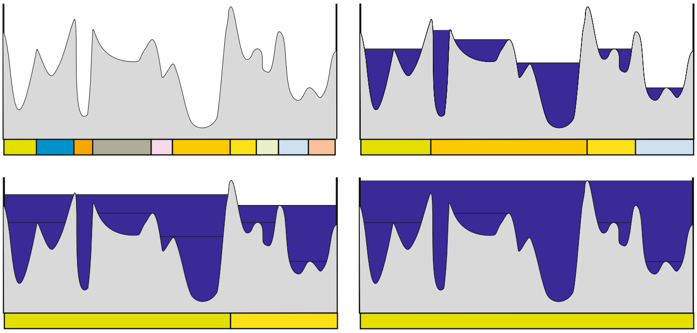


Fig. 1. 4 levels of the waterfall hierarchy. The watershed partition is presented below each topographic surface. Each topographic surface has been flooded up to the lowest pass point of the catchment basins associated to the preceding surface.

After a rapid discussion of the classical constructions of the waterfall hierarchy, we present two new methods having each its merits.

The first method models the topographic surface as a particular graph, perfectly representing the waterfall flooding, where all lakes are full, at the verge of overflow. The nodes of the graph represent the full lakes and their weight equals the altitude of the lake. A drop of water added to a full lake provokes an overflow into a neighboring lake, called exhaust lake of the first one. An edge links each lake with its exhaust lakes ; its weight is equal to the altitude of the lake which overflows through this edge. Such node and edge weighted graphs are called flooding graphs. On such a graph the trajectory of a drop of water is the same, whether one considers the edge weights alone or the node weights alone. Efficient, precise and fast algorithms based on the node weights only permit to construct the watershed partitions [16,17,18].

The second method constructs a complete waterfall hierarchy in one pass through the edges of the minimum spanning tree of the region adjacency graph

2 The Waterfall Hierarchy: State of the Art

2.1 An Algorithm Based on Successive Image Floodings

The waterfall hierarchy has been defined by Serge Beucher for grey tone images [3]. A first algorithm works at the image level [4]. Each catchment basin is flooded up to its lowest pass point. As the pass points are regional minima of the watershed line, the solution consists in the following steps:

* constructing the watershed line of the topographic surface f, producing the level 1 of the waterfall hierarchy.

* constructing a ceiling function equal to f on the watershed line and ∞ elsewhere

* constructing the highest flooding of f under this ceiling function

A new topographic surface is produced, which is then submitted to the same process, producing the next level of the waterfall hierarchy.

Discussion: Using the watershed line is an advantage, as the ceiling function is easily constructed. It also is a disadvantage, as the watershed line has no meaning in terms of image content. It "takes place" to the detriment of thin and narrow structures which cannot be represented. It makes the construction of a hierarchy more difficult. Below we present a method where each level of the waterfall hierarchy is a partition, without watershed line separating neighboring tiles.

2.2 A Watershed Algorithm on the Region Adjacency Graph

A second algorithm [5] constructs the catchment basins of the topographic surface at the pixel level. They constitute the finest level of the waterfall hierarchy. The higher levels of the waterfall hierarchy are then constructed on the region adjacency graph, whose nodes are the catchment basins, whose edges represent the pass points between neighboring basins, weighted by the altitude of the pass point. B.Marcotegui et al use the watershed partition of this graph by constructing a minimum spanning forest rooted in its regional minima [13]. The edge inside each tree are then contracted within the MST, producing a new tree on which the same process is repeated. This has been the first watershed algorithm on edge weighted graphs. More recent alternative algorithms could be used for the construction of the watershed on edge weighted graphs [8,10,17].

Discussion: The minimum spanning tree of the region adjacency graph is not unique and with a unlucky choice, the results could be poor. This is due to the fact that the watershed algorithms on an edge weighted graph are myopic and evaluate the steepness of the flooding on the immediate neighboring edges of each node. Below we present a watershed algorithm in which the steepness of the flooding is better taken into account.

3 Construction of the Waterfall Hierarchy, Thanks to the Flooding Graph

3.1 Reminders on Node and/or Edge 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, π, is a sequence of vertices and edges, interweaved in the following way: π 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.

Edges and/or nodes may be weighted. Denote by $\mathcal{F}_e$ and $\mathcal{F}_n$ the sets of non negative weight functions on the edges and on the nodes respectively. The function $\eta \in \mathcal{F}_e$ takes the value η_{pq} on the edge e_{pq}, and the graph holding only edge weights is designated by $G(\eta, nil)$. The function $\nu \in \mathcal{F}_n$ takes the weight

ν_p on the node p and the graph holding only these node weights is designated by $G(nil, \nu)$. If both nodes and edges are weighted we write $G(\eta, \nu)$.

Images defined on a grid may also be modelled by such a graph : the pixels of the image become the nodes of the graph, with the same weight ; neighboring pixels are linked by an unweighted edge.

We define two operators between the node and edge weights:

* an operator δ_{en} associating to the function $\nu \in \mathcal{F}_n$ the function $\delta_{en}\nu \in \mathcal{F}_e$ taking the value $(\delta_{en}\nu)_{pq} = \nu_p \vee \nu_q$ on the edge e_{pq}.
* an operator ε_{ne} associating to the function $\eta \in \mathcal{F}_e$ the function $\varepsilon_{ne}\eta \in \mathcal{F}_n$ taking the value $(\varepsilon_{ne}\eta)_p = \bigwedge_{(s\,\text{neighbors of}\,p)} \eta_{ps}$ on the node p.

The pair of operators $(\delta_{en}, \varepsilon_{ne})$ form an adjunction:

$\forall \eta \in \mathcal{F}_e, \forall \nu \in \mathcal{F}_n : \delta_{en}\nu < \eta \Leftrightarrow \nu < \varepsilon_{ne}\eta$. It follows that ([21]):
* δ_{en} is a dilation from $\mathcal{F}_n$ into $\mathcal{F}_e$
* ε_{ne} is an erosion from $\mathcal{F}_e$ into $\mathcal{F}_n$

This section is based on the flooding graph, whose nodes represent the full lakes of a topographic surface when the flooding of each basin reaches its lowest pass points . We first rigorously establish the altitude of a full lake. The flooding graph is completed by adding the edges linking each lake with its exhaust lakes. The flooding graph contains the necessary and sufficient information for constructing the watershed partition or the waterfall hierarchy of a topographic surface.

3.2 The Lowest Pass Point of a Catchment Basin

Consider a watershed partition π of a topographic surface, verifying the following condition : if s and p are two neighboring nodes belonging to the catchment basins of two distinct minima m_1 and m_2, then there exists a non increasing path going from s to m_1 and a non increasing path from p to m_2.

Our aim is to figure out the level of the full lakes on such a topographic surface, given this watershed partition.

Consider a catchment basin CB_1 filled by a lake up to its lowest pass point, leading to an unflooded neighboring basin CB_2. The regional minima of CB_1 and CB_2 are respectively m_1 and m_2.

Consider among all paths linking the minimum m_1 with the minimum $m_{2,}$ a path whose highest altitude is the lowest. Such a path crosses the pass point between both basins CB_1 and CB_2. The highest altitude of the path is equal to the altitude of the pass point.

Consider the altitude of the flooded relief g along one such path ϖ (there may be several of them). The altitude is constant and equal to the altitude of the full lake covering m_1 until it reaches the last node with this altitude, say p ; the next node q having a lower altitude. But $g_p > g_q \Rightarrow g_p = f_p$, an implication which characterizes floodings [14] indicating that the node p is dry and has the altitude of the full lake. Three cases are to be considered:

* $p \in CB_1$ and $q \in CB_2$. Then $f_p = g_p > g_q \geq f_q$ and the altitude of the lake is $f_p \vee f_q = f_p$

* $p \in CB_2$ and $q \in CB_2$ and the last node preceding p on the path ϖ is a node s belonging to CB_1, then $f_s \leq g_s = g_p = f_p$. The altitude of the lake is $f_s \vee f_p = f_p$

* $p \in CB_2$ and $q \in CB_2$ and the last pixel of the path ϖ belonging to CB_1 is a pixel s, followed by a series of pixels $t_1, t_2, ...p$ belonging to CB_2. All these pixels belong to the full lake : $g_s = g_{t_1} = ... = g_p$. On the other hand $t_1, t_2, ...p$ constitutes a non ascending path for the function f : $f_{t_1} \geq f_{t_1} \geq ... \geq f_p$. As $g_p = g_{t_1} \geq f_{t_1} \geq f_p = g_p$, all internal inequalities are in fact equalities and $f_{t_1} = g_p$. Here again the level of the lake is $f_s \vee f_{t_1}$ for two pixels belonging one to CB_1 and the other to CB_2.

This gives a clue for determining the altitude of the full lake which will cover the minimum m_1. One considers all couples of neighboring pixels which form the boundary of CB_1, with $p \in CB_1$ and $q \notin CB_1$. The altitude of the full lake is then the smallest value taken by $f_p \vee f_q$: $\lambda(CB_1) = \bigwedge_{\substack{p \in CB_1 \text{ and } q \notin CB_1 \\ p,q \text{ neighbors}}} f_p \vee f_q$.

Each exhaust passpoint of the catchment basin $CB1$ towards a basin $CB2$ is thus a pair of nodes $p \in CB_1$ and $q \in CB_2$ verifying $f_p \vee f_q = \lambda(CB_1)$. We say that $CB2$ is an exhaust basin of $CB1$, as each additional drop of water falling in the full lake occupying $CB1$ provokes an overflow into $CB2$.

3.3 Modelling the Catchment Basins of a Topographic Surface as a Flooding Graph

Modeling a Topographic Surface as a Flooding Graph. Consider a topographic surface in which each catchment basin has been flooded up to its exhaust pass point. We create a graph $G = [N, E]$ in which each node n_i represents a full lake occupying a catchment basin, with a weight ν_i equal to the altitude of this lake. An edge e_{ij} with the same weight $\eta_{ij} = \nu_i$ links this node n_i with the node n_j if the catchment basin CB_j is an exhaust catchment basin of CB_i. We say that e_{ij} is an exhaust edge of n_i. We call this node and edge weighted graph a flooding graph of f, as it models the highest flooding of f without overflow.

Properties of a flooding graph. As each catchment basin has one or several lowest pass points, each node n_k of $G = [N, E]$ has one or several exhaust edges. All other adjacent edges of n_k correspond to higher pass points of CB_k and represent the exhaust edges of neighboring catchment basins. Hence each exhaust edge of a node has the lowest weight among all adjacent edges of this node: $\nu = \varepsilon_{ne}\eta$. In particular, if a node n_i is linked with n_k through an exhaust edge e_{ik} of n_k, we have $\nu_k = \eta_{ik} \geq \nu_i$. It follows that each node is linked by an exhaust edge with its neighboring nodes whose weight is lower or equal ; and the weight of each edge is equal to the highest weight of its extremities : $\eta_{ik} = \nu_i \vee \nu_k$. Hence $\eta = \delta_{en}\nu$.

A graph where the node and edge weights are coupled by $\nu = \varepsilon_{ne}\eta$ and $\eta = \delta_{en}\nu$ is called a flooding graph [18].

Flooding paths and regional minima. Adding a drop of water to a full lake provokes an overflow through an exhaust pass point into a neighboring full lake with a lower or equal altitude, which itself provokes an overflow and so on until, ultimately, a regional minimum lake is reached. Such a trajectory of a drop of water is modelled on the flooding graph as a flooding path.

Definition 1. *A path $p - e_{pq} - q - e_{qs} - s...$ is a flooding path if each node except the last one is followed by one of its exhaust edges.*

By construction we then have $\nu_p = \eta_{pq} \geq \nu_q = \eta_{qs} \geq \nu_s$. Considering only the node weights, the path will be a flooding path if $\nu_p \geq \nu_q \geq \nu_s$. Considering only the edge weights it is a flooding path if each edge is an exhaust edge of the preceding node. A drop of water following a flooding path ultimately is trapped in a regional minimum. It arrives to a node whose exhaust lead to adjacent nodes with the same weight. There is no possibility to reach a deeper node as each neighboring node of a regional minimum has a higher weight and is linked with the regional minimum by an exhaust edge also with a higher weight.

The catchment zone of a regional minimum is the set of nodes linked with this minimum through a flooding path. Minimum and flooding path being the same whether one considers the node weights only or the edge weights only.

Having the same flooding paths and the same regional minima, both graphs also have the same catchment zones.

Theorem 1. *For a flooding graph $G(\eta, \nu)$, the flooding paths, regional minima and catchment zones of the graphs $G(nil, \nu)$ and $G(\eta, nil)$ are identical.*

An Order Relation Between Flooding Paths. A drop of water following a flooding path reaching a regional minimum node p can quit this node through one of its exhaust edges and reach another node q with the same altitude. From this node, it may then come back. Going hence and forth between p and q, the flooding path is thus artificially prolongated as a path of infinite length. 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.

Constructing the Steepest Watershed Partition. 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.

The classical algorithm for constructing a watershed partition in which the flooding of a node weighted graph is governed by a hierarchical queue [12] propagates the labels of the minima along the steepest flooding paths of the graph. We start detecting and labeling the node regional minima.

Initialisation: Create a hierarchical queue HQ. Put the outer boundary nodes of the regional minima in the queue corresponding to their weight. Assign to each regional minimum a distinct label

```
Repeat until the HQ is empty:
Extract the node $p$ with the highest priority from the queue.
    For each node $q$ without label, such that $(p, q)$ neighbors:
        * $label(q) = label(p)$
        * put $q$ in the queue with priority $n_q$
```

At the end of the process, we obtain a watershed partition Π_1 of the nodes. By merging all basins corresponding to nodes belonging to the same tile of Π_1, we obtain a new partition of the domain D. The boundaries of this partition are a subset of the boundaries of the initial watershed partition of f.

The complete process is repeated: creating the flooding graph of the new partition, constructing the watershed partition Π_2 on the nodes etc.

The Flooding Graph of a Region Adjacency Graph. An alternative solution consists in constructing the region neighborhood graph $G = [N, E]$ associated to the initial watershed partition. The subsequent processing is done on this graph and illustrated in fig.2. The flooding graph G' is constructed first: for each node p of G, one retains in G' only its exhauts edges, i.e. the adjacent edges with minimal weight. The node p gets the same weight as its exhaust edges. The watershed partition of G' is constructed as above, producing a number of connected components. Contracting the inside edges of each component within the graph G, produces a new edge weighted graph, to which the same process may be applied.

4 The Waterfall Partition in One Run

We now compute the waterfall hierarchy associated to an arbitrary edge weighted tree T with edge weights $\eta > 0$. In the context of morphological segmentation, T is the minimum spanning tree of the region adjacency graph associated to the topographic surface. But the developments below make sense for any edge weighted tree. The waterfall hierarchy will be expressed as another tree Θ, having the same structure, i.e. nodes and edges as the tree T but not the same weights. We associate to the tree Θ the ultrametric distance χ : the distance between two nodes n_1 and n_2 is equal to the highest edge on the unique path inside Θ linking these two nodes.

Initially all edges are assigned a weight equal to ∞. Hence the ultrametric distance between any two node also is equal to ∞.

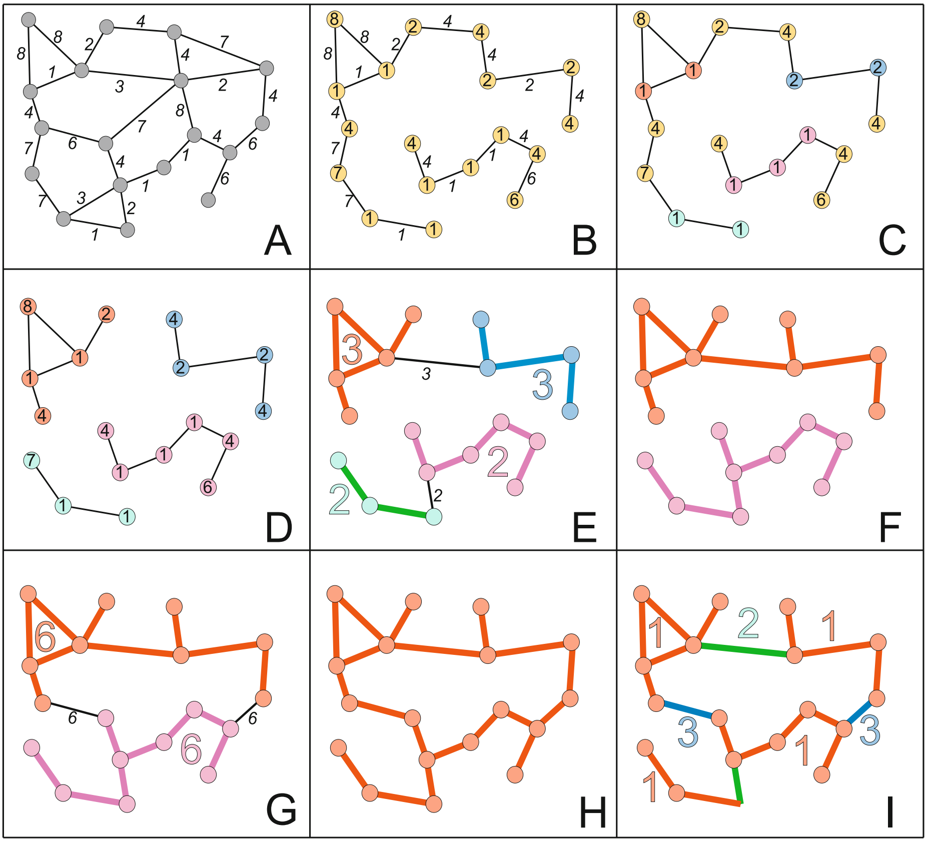


Fig. 2. A: the region adjacency edge weighted graph G ; B: the flooding graph, by retaining for each node its exhaust edges ; C: assigning to each node the weight of its exhaust edges and skipping the edge weights ; D: watershed partition on the nodes obtained by the hierarchical queue algorithm ; E: contraction of all edges which belong to the same tile of the watershed partition These contracted edges are indicated as bold lines. At the same time, construction of the associated flooding graph: one retains for each node obtained by contraction its lowest adjacent edges within the graph G. Each node gets the weight of its exhaust edge ; F: the watershed partition associated to the previous node weighted graph ; G: new contraction of edges, and associated flooding graph. Each node gets the weight of its exhaust edge. There is only one regional minimum ; H: the associated watershed partition has only one region, which marks the end of the process ; I: the waterfall hierarchy of all edges which have been used as exhaust edges : each edge is indexed by the time when it has been introduced in a flooding graph.

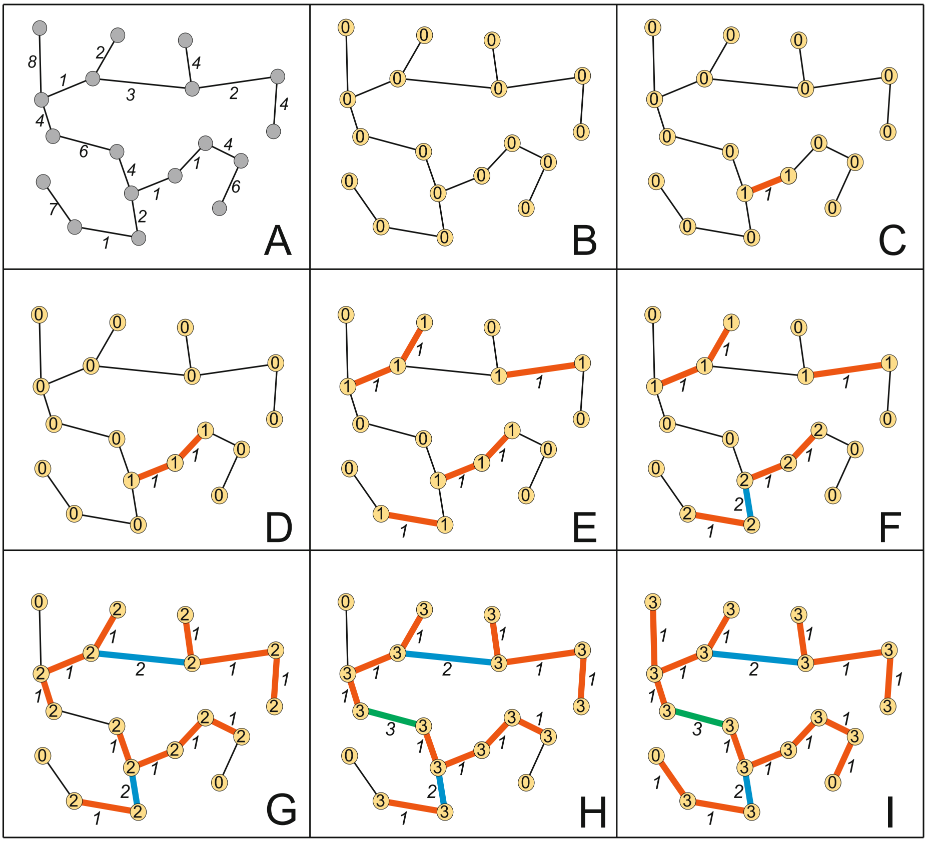


Fig. 3. A: Minimum spanning tree ; B: Initially Θ has only nodes, with a diameter 0 ; C: The first edge is introduced in Θ with a weight 1. Its extremities have a weight 1 ; D,E: Additional edges are introduced in Θ with a weight 1. Their extremities have a weight 1 ; F: First edge linking two subtrees of Θ with diameter 1. This edge gets a waterfall weight 2, and the diameter of its subtree is 1 ; G: Introduction of new edges with a waterfall level of 1. The diameters of the subtrees is 2 ; H: Introduction of an edge linking two subtrees with a diameter 2. The resulting edge gets a weight equal to 3 and its subtree a diameter 3 ; I: After introduction of the last edge, all edges got their waterfall index.

We define the open ball $\overset{\circ}{\mathrm{Ball}}(p,\lambda) = \{q \mid \chi(p,q) < \lambda\}$ and $\mathrm{diam}(p)$ its diameter, i.e. the weight of the highest edge linking two nodes of $\overset{\circ}{\mathrm{Ball}}(p,\infty)$. Initially no such edge is present and we set $\mathrm{diam}(p) = 0$.

We start with the minimum spanning tree T of the neighborhood graph. We treat the edges in ascending order of their weights η. Let $u = (p,q)$ the current edge to be processed:

- $\Theta = \Theta \cup \{u\}$
- $\theta(u) = 1 + (\mathrm{diam}(p) \wedge \mathrm{diam}(q))$

Analysis of the Algorithm (Illustration in Fig.3). Before adding the edge u to Θ, $\overset{\circ}{\mathrm{Ball}}(p,\infty)$ and $\overset{\circ}{\mathrm{Ball}}(q,\infty)$ are disjoint, having respectively the diameters d_p and d_q. After adding the edge u to Θ with the weight $\theta(u)$, $\overset{\circ}{\mathrm{Ball}}(p,\infty) = \overset{\circ}{\mathrm{Ball}}(q,\infty)$ and the largest edge linking two nodes is either included in one of the previous balls $\overset{\circ}{\mathrm{Ball}}(p,\infty)$ or $\overset{\circ}{\mathrm{Ball}}(p,\infty)$ or it is the edge u itself. Hence the new diameter of $\overset{\circ}{\mathrm{Ball}}(p,\infty) = \overset{\circ}{\mathrm{Ball}}(q,\infty)$ is equal to $\theta(u) \vee \mathrm{diam}(p) \vee \mathrm{diam}(q)$.

Let us analyze $\theta(u) = 1 + (\mathrm{diam}(p) \wedge \mathrm{diam}(q))$:

- if for instance $\mathrm{diam}(p) < \mathrm{diam}(q)$ then $\mathrm{diam}(p) \leq \mathrm{diam}(q) - 1$ and $1 + (\mathrm{diam}(p) \wedge \mathrm{diam}(q)) = 1 + \mathrm{diam}(p) \leq \mathrm{diam}(q)$. The adjunction of the edge u does not change the diameter of $\overset{\circ}{\mathrm{Ball}}(q,\infty)$; it increases the diameter of $\overset{\circ}{\mathrm{Ball}}(p,\infty)$ by 1.
- if $\mathrm{diam}(p) = \mathrm{diam}(q)$, then $\theta(u) = 1 + \mathrm{diam}(p) = 1 + \mathrm{diam}(q)$ and the adjunction of the edge u increases the diameter of $\overset{\circ}{\mathrm{Ball}}(p,\infty)$ and of $\overset{\circ}{\mathrm{Ball}}(q,\infty)$ by 1.

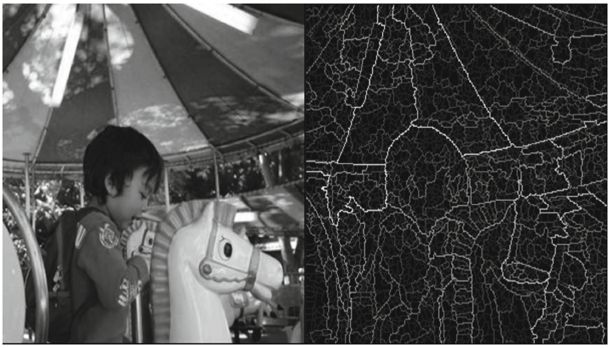
5 Discussion

Two methods have been presented for constructing the waterfall hierarchy of a topographic surface. The second is the fastest but relies upon a number of arbitrary choices: choice of a minimum tree among many of the regiona adjacency graph. The edges of the tree are processed in the order of increasing values ; among edges with the same weight, there are again many arbitrary choices. The structure of the waterfall hierarchy will remain the same, but the boundaries of the partitions forming each level will differ. The first method is the most precise as it relies on a watershed algorithm which reduces to the outmost the arbitrary choices between alternative solutions [16,17,18] For instance, if the regional minima have distinct weights, the watershed is unique .

6 Conclusion

The waterfall hierarchy of a grey tone image is illustrated by the waterfall saliency of its contours in figure <ref>theowfal</ref>.

The waterfall hierarchy of a grey tone image is illustrated by the waterfall saliency of its contours in the following figure.



Initial image and the saliency of its waterfall contours

The waterfall hierarchy permits a structural analysis of a topographic surface as it explores the nested structure of the catchment basins, without any consideration of depth or size. It is in fact a purely ordinal process, as shown by the last algorithm presented above, which increases by one the waterfall level each time one encounters a deeper nested level.

The waterfall hierarchy has been invented, with floodings on a topographic surface in mind. It has then be, conceptually and algorithmically, extended to node or edge weighted graphs. It may be extremely useful for structuring large graphs representing complex interconnections in many domains, beyond the world of image processing.

Sometimes, there are only very few nested structures, and one may wish enriching the watershed hierarchy. B.Marcotegui and S.Beucher have shown how to prevent some regions from mergings. This domain of research merits further investigations.

References

1. Angulo, J., Serra, J.: Color segmentation by ordered mergings. In: Image Processing, ICIP 2003, vol. 2, pp. 121–125. IEEE (2003)
2. Angulo, J., Velasco-Forero, S., Chanussot, J.: Multiscale stochastic watershed for unsupervised hyperspectral image segmentation. In: Geoscience and Remote Sensing Symposium, IGARSS 2009, vol. 3, pp. 93–96. IEEE (2009)
3. Beucher, S.: Segmentation d'Images et Morphologie Mathématique. PhD thesis, E.N.S. des Mines de Paris (1990)
4. Beucher, S.: Watershed, hierarchical segmentation and waterfall algorithm. In: Mathematical Morphology and its Applications to Image Processing, pp. 69–76. Springer (1994)
5. Marcotegui, B., Beucher, S.: Fast implementation of waterfalls based on graphs. In: ISMM 2005: Mathematical Morphology and its Applications to Signal Processing, pp. 177–186 (2005)

6. Beucher, S.: Geodesic reconstruction, saddle zones and hierarchical segmentation. Image Anal. Stereol. 20(2), 137–141 (2001)
7. Beucher, S., Marcotegui, B., et al.: P algorithm, a dramatic enhancement of the waterfall transformation (2009)
8. Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed cuts: Minimum spanning forests and the drop of water principle. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 1362–1374 (2009)
9. Delest, S., Bone, R., Cardot, H.: Fast segmentation of triangular meshes using waterfall. In: VIIP, vol. 6, pp. 308–312 (2006)
10. Golodetz, S.M., Nicholls, C., Voiculescu, I.D., Cameron, S.A.: Two tree-based methods for the waterfall. Pattern Recognition 47(10), 3276–3292 (2014)
11. Hanbury, A., Marcotegui, B.: Waterfall segmentation of complex scenes. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3851, pp. 888–897. Springer, Heidelberg (2006)
12. Meyer, F.: Un algorithme optimal de ligne de partage des eaux. Actes duu, 847–859 (1991)
13. Meyer, F.: Minimum spanning forests for morphological segmentation. In: Procs. of the Second International Conference on Mathematical Morphology and its Applications to Image Processing, pp. 77–84 (1994)
14. Meyer, F.: Flooding and segmentation. In: Goutsias, J., Vincent, L., Bloomberg, D.S. (eds.) Mathematical Morphology and its Applications to Image and Signal Processing. Computational Imaging and Vision, vol. 18, pp. 189–198 (2002)
15. Meyer, F., Stawiaski, J.: Morphology on graphs and minimum spanning trees. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) ISMM 2009. LNCS, vol. 5720, pp. 161–170. Springer, Heidelberg (2009)
16. Meyer, F.: The steepest watershed: from graphs to images. arXiv preprint arXiv:1204.2134 (2012)
17. Meyer, F.: Watersheds, waterfalls, on edge or node weighted graphs. arXiv preprint arXiv:1204.2837 (2012)
18. Meyer, F.: Watersheds on weighted graphs. Pattern Recognition Letters (2014)
19. Ogor, B., Haese-coat, V., Ronsin, J.: Sar image segmentation by mathematical morphology and texture analysis. In: Geoscience and Remote Sensing Symposium, IGARSS 1996. 'Remote Sensing for a Sustainable Future.', International, pp. 1:717–719 (1996)
20. Shafarenko, L., Petrou, M., Kittler, J.: Automatic watershed segmentation of randomly textured color images. IEEE Transactions on Image Processing 6(11), 1530–1544 (1997)
21. Serra, J. (ed.): Image Analysis and Mathematical Morphology. II: Theoretical Advances. Academic Press, London (1988)
22. Stottinger, J., Banova, J., Ponitz, T., Sebe, N., Hanbury, A.: Translating journalists' requirements into features for image search. In: 15th International Conference on Virtual Systems and Multimedia, VSMM 2009, pp. 149–153. IEEE (2009)