Watershed Algorithms and Contrast Preservation

Laurent Najman and Michel Couprie

Laboratoire A2SI, Groupe ESIEE
Cité Descartes, BP99
93162 Noisy-le-Grand Cedex France
{l.najman,m.couprie}@esiee.fr
http://www.esiee.fr/~coupriem/Sdi/

Abstract. This paper is devoted to the study of watershed algorithms behavior. Through the introduction of a concept of *pass value*, we show that most classical watershed algorithms do not allow the retrieval of some important topological features of the image (in particular, saddle points are not correctly computed). An important consequence of this result is that it is not possible to compute sound measures such as depth, area or volume of basins using most classical watershed algorithms. Only one watershed principle, called *topological watershed*, produces correct watershed contours.

Keywords: Mathematical Morphology, Watersheds, Contours Saliency, Topology

1 Introduction

This paper is a first of a series dedicated to the notion of watershed contour saliency. Using this concept, introduced in [1,2], we can sum up in one image all the contour information that we can obtain by filtering the image by attribute opening [3,4,5,6] for all values of the parameter and applying a watershed on each of the filtered images. Several algorithms [2,7,8,9] for computing saliency of watershed contours have been proposed. We expect to obtain the same result either by thresholding the saliency image at a given level k, or by filtering the original image using an attribute opening using k as parameter value and applying a watershed algorithm on the filtered image. None of the existing saliency algorithms computes this expected result. The goal of the series is to show why it is the case, and to propose a novel efficient algorithm that computes the expected result.

This paper is devoted to the study of watershed algorithms behavior with respect to what is needed to compute saliency of contours. For computing saliency, one needs a map of watershed basins neighborhood, with the altitude of their associated saddle points, and a valuation on each basin. A review of watershed algorithms and their associated results can be found in [10]. This review does not study algorithms from the point of view of the preservation of important

topological features of the original image; for instance, it does not consider the question: does the algorithm compute correct saddle points? We are going to tackle the difficult notion of saddle point through the introduction of a concept of "pass value". We demonstrate that watershed algorithms that are the most used in practice do not behave correctly with respect to the preservation of pass values, and thus cannot be used in a saliency algorithm. We show that the approach called *topological watershed* [11] (which is not mentioned in [10]) provides the only existing algorithm that produces a correct entry point for a saliency algorithm.

2   Brief Description of Watershed Algorithms

2.1   Intuitive Notions for Watershed

The intuitive idea underlying the watershed notion comes from the field of topography: a drop of water falling on a relief follows a descending path and eventually reaches a minimum. Watershed lines are the divide lines of the domains of attraction of drops of water. This intuitive approach is not well suited to practical implementations, and can yield biased results in some cases [12]. An alternative approach is to imagine the surface being immersed in a lake, with holes pierced in local minima. Water will fill up basins starting at these local minima, and, at points where waters coming from different basins would meet, dams are built. As a result, the surface is partitioned into regions or basins separated by dams, called watershed lines.

2.2   What Is a Watershed Algorithm

This paper is not the place to describe in details the (large) family of watershed algorithms. Nevertheless, it is worthwhile to give a brief description of the main algorithms. Let E be a set of vertices (or points). Let $\mathcal{P}(E)$ denote the set of all subsets of E. Let $G = (E, \Gamma)$ be a (symmetric) graph, where Γ is a mapping from E into $\mathcal{P}(E)$, which associates to each point x of E, the set $\Gamma(x)$ of points adjacent to x.

Let $X \subseteq E$, and let $x_0, x_n \in X$. A *path* from x_0 to x_n in X is an ordered family $(x_0, x_1, \ldots, x_n)$ of points of X such that $x_{i+1} \in \Gamma(x_i)$, with $i = 0 \ldots n-1$. Let $x, y \in X$, we say that *x is connected to y* if there exists a path from x to y in X. The relation "is connected to" is an equivalence relation. A *connected component* of X is an equivalence class for the relation "is connected to".

Let $E = \mathcal{Z}^2$. We denote by $\mathcal{F}(E)$ the set composed of all functions from E to $\mathcal{Z}$. Let $F \in \mathcal{F}(E)$. We denote by $\overline{X}$ the complement of X. We write $F_k = \{x \in E; F(x) \geq k\}$ with $k \in \mathcal{Z}$; F_k is called an *upper (cross-) section* of F, and $\overline{F_k}$ is called a *lower (cross-) section* of F. A connected component of a section $\overline{F_k}$ is called a *(level k) lower-component* of F. A level k lower-component of F that does not contain a level $(k-1)$ lower-component of F is called a *(regional) minimum* of F.

Let us recall that a *partition* of a set S is a collection of non-empty disjoint subsets of S whose union is S.

A *watershed algorithm* builds a partition of the space:

- it associates an influence zone $B(M)$ called *catchment basin*, to each minimum M of the image. The set $B(M)$ is connected and contains M;
- it may produce a set of *watershed lines* which separates those catchment basins one from each other.

2.3 Vincent-Soille Watershed Algorithm [12]

For any set A and any set $B \subset A$ made of several connected components B_i, the *geodesic influence zone* $iz_A(B_i)$ *of* B_i *in* A is the locus of the points of A whose geodesic distance to B_i is strictly smaller than their geodesic distance to any other component of B. We define the following recursion:

$$X_{h_{\min}+1} = \overline{F_{h_{\min}+1}} = MIN_{h_{\min}} \tag{1}$$
$$X_{h+1} = MIN_h \cup IZ_{\overline{F_{h+1}}}(X_h) \tag{2}$$

where $h_{\min}$ is the lowest grey-value of F, where $IZ_{\overline{F_{h+1}}}(X_h)$ is the union of the geodesic influence zones of the connected components of X_h in $\overline{F_{h+1}}$, and where MIN_h is the union of minima of F with grey-level equal to h. The watershed lines are the complement of $X_{h_{\max}+1}$.

As noted in [10], Vincent-Soille's algorithm does not implement exactly this recursion. Thanks to a fifo queue, it floods the catchment basins of the image, and to build the watershed lines, it associates a special value WSHED to the pixels where two different cacthment basins would merge. A point labelled WSHED by the algorithm is not considered again in the following iteration, as it should be the case. Furthermore, pixels labelled WSHED are propagated. This allows the detection of special thick watershed zones, like those called buttonholes (see fig. 2.a).

2.4 Meyer's Watershed Algorithm [13]

Starting from a greyscale image F and a set M of markers with different labels (in our case, these will be the minima of F), it expands as much as possible the set M, while preserving the number of connected components of M:

1. insert every neighbor x of every marked area in a hierarchical queue, with a priority level corresponding to the grey level $F(x)$. Note that a point cannot be inserted twice in the queue;
2. extract a point x from the hierarchical queue, at the highest priority level, that is, the lowest grey level. If the neighborhood $\Gamma(x)$ of x contains only points with the same label, then x is marked with this label, and its neighbors that are not yet marked are put into the hierarchical queue;

Step 2 must be repeated until the hierarchical queue is empty. The watershed lines set is the complement of the set of labeled points. Let us note that this algorithm does neither label nor propagate watershed pixels, which "stop" the flooding. Thus, the watershed lines produced by Meyer's algorithm are always thinner than lines produced by other watershed algorithms.

2.5   Cost-Based Watershed

The principle is to define a distance or a cost for travelling between pixels, and to define the influence zone of a minimum as the set of points which are strictly closer to this minimum than to any other minimum. Various costs or distances can be considered, the most popular ones being the *topographical distance* [14,15], but other approaches exist [16] among which we can mention the max-arc path cost.

The (so-called) topographical distance of an image F is a digital analogue to $dt_F(x,y) = \inf_{\pi \in \Pi(x,y)} \int_\pi ||\nabla F(\pi(s))||ds$. Let us note that if we are on a line of steepest slope between x and y, then $dt_F(x,y) = |F(x) - F(y)|$. The catchment basin of a minimum m_i is defined as the set of pixels x for which $F(m_i) + dt_F(m_i, x) < F(m_j) + dt_F(m_j, x)$ for all minima $m_j \neq m_i$. The watershed lines set is the complementary of those catchment basins.

Another simple possible choice is the *max-arc path cost* [16] which assigns to a path the maximum of F for each pixel on the path. In this case, $dm_F(x,y) = \inf_{\pi \in \Pi(x,y)} \max_i F(\pi(i))$, and the catchment basin of a minimum m_i is defined as the set of pixels x for which $dm_F(m_i, x) < dm_F(m_j, x)$ for all minima $m_j \neq m_i$. The watershed lines set is the complementary of those catchment basins.

2.6   Topological Watershed [11]

The idea is to define a transform that acts directly on the greyscale image, by lowering some points in such a manner that the connectivity of each lower cross-section $\overline{F_k}$ is preserved. The regional minima of the result, which have been spread by this transform, can be interpreted as the catchment basins. The formal definition relies on a particular notion of simple point:

Definition 1. *Let $G = (E, \Gamma)$ be a graph, and let $X \subset E$.*
The point $x \in X$ is simple (for $\overline{X}$) if the number of connected components of $\overline{X} \cup \{x\}$ equals the number of connected components of $\overline{X}$. In other words, x is simple (for $\overline{X}$) if x is adjacent to exactly one connected component of $\overline{X}$.

We can now define a notion of destructible point, and the topological watershed:

Definition 2. *Let $F \in \mathcal{F}(E)$, $x \in E$, and $k = F(x)$. The point x is destructible (for F) if x is simple for $\overline{F_k}$.*
We say that $W \in \mathcal{F}(E)$ is a topological watershed of F if W may be derived from F by iteratively lowering destructible points by one until stability (that is, until all points of E be non-destructible for W).

The catchment basins *of the topological watershed W are the minima of W, and the* watershed lines *are the non-minima of W.*

As a consequence of this definition, a topological watershed W of a function F is a function which has the same number of regional minima as F. Furthermore, the connectivity of any lower cross-section is preserved during this transformation.

Let us note that, in this case, and contrary to other watershed principles, the watershed lines are part of the definition: it does not exist a variation of this notion that does not build those lines. An efficient algorithm to compute the topological watershed has been proposed in [11].

Let us emphasize the essential difference between this notion of topological watershed and the notion of homotopic greyscale skeleton, pioneered by Goetcherian [17] and extensively studied in [18,19]. With the topological watershed, only the connected components of the lower cross-sections of the function are preserved, while the homotopic greyscale skeleton preserves both these components and the components of the upper cross-sections. As a consequence, an homotopic greyscale skeleton may be computed by using a purely local criterion for testing whether a point may be lowered or not, while computing a topological watershed requires the use of a global data structure [11].

3   Watershed Algorithms Comparison

Intuitively, for application to image analysis, the watershed lines represent the location of pixels which best separate the dark objects (regional minima), in terms of grey level difference (contrast). In order to evaluate the effectiveness of this separation, we have to consider the values of pixels along watershed lines. This motivates the following definition.

Definition 3. *The* watershed contours *of F is a grayscale image W such that $W(x) = 0$ for any x in a catchment basin, and $W(x) = F(x)$ elsewhere.*

Let us note that such a definition is not necessary for the topological watershed, which produces a function, and not a binary result.

3.1   Saddle Point, Pass Value and the Dynamics

To formalize the notion of contrast between two minima, we need to characterize first-contact points between basins. In the continuous framework, such points are called saddle points, but this notion is difficult to transfer to the digital grid. Furthemore, such a notion is not fundamental for contrast criteria. More precisely, for each couple of neighboring basins, we only need the altitude of the lowest contact point between them. This is the motivation for defining the pass value, a natural concept already used by several authors.

Definition 4. *Let $\mathcal{M}(F)$ be the set of all minima of F. We define the* pass value *$F(m_1, m_2)$ between two minima m_1 and m_2 in $\mathcal{M}(F)$ as*

$$F(m_1, m_2) = \min_{\pi \in \Pi(m_1, m_2)} \max_i F(\pi(i)) \tag{3}$$

where $\Pi(m_1, m_2)$ is the set of all paths linking m_1 to m_2.

For applications to image analysis like filtering, and especially for saliency, we want to compute

- all pass values; we would like watershed contours to have the same pass values as the original image;
- and a measure of contrast or importance of each basin (minima) of the original image; such a measure should correspond to measure taken on lower cross-sections of the original image.

Various contrast measures can be computed, among which we can mention depth (dynamics [20]), area and volume [3]. We are going to examine more particularly the case of the dynamics.

We first recall the basic definitions introduced by Grimaud [20] (in fact these definitions were proposed for 2D images, we extend them for arbitrary graphs). Let $F \in \mathcal{F}(E)$ and let X be a minimum for F. The *attraction domain of X* is the set composed of all point x such that there exist a descending path from x to X. The attraction domain of a minimum X is denoted by $K(X)$.

Let π be a path. The *dynamics of π (for F)* is the value $Dyn(\pi) = Max\{|F(x) - F(y)|;$ for all x, y in $\pi\}$.

Let x, y be two points. The *dynamics between x and y (for F)* is the value $Dyn(x, y) = Min\{Dyn(\pi);$ for all $\pi \in \Pi(x, y)\}$.

Let X and Y be two subsets of E. The *dynamics between X and Y (for F)* is the value $Dyn(X, Y) = Min\{Dyn(x, y);$ for all $x \in X$, $y \in Y\}$.

Definition 5. *Let $X \in \mathcal{M}(F)$. The* dynamics of X (for F) *is the number $Dyn(X)$ such that:*

- *If $F(X) = Min\{F(Y); Y \in \mathcal{M}(F)\}$, then $Dyn(X) = \infty$;*
- *Otherwise, $Dyn(X) = Min\{Dyn[X, K(Y)]; \forall Y \in \mathcal{M}(F), F(Y) < F(X)\}$.*

3.2   The Case of the Topological Watershed

We can prove [21] that the topological watershed preserves the pass values.

Property 1 *Let W be a topological watershed of F. For all $(m_1, m_2) \in \mathcal{M}(F)^2$, and for the corresponding minima $(m'_1, m'_2) \in \mathcal{M}(W)^2$, we have*
$$F(m_1, m_2) = W(m'_1, m'_2)$$

In the sequel of the paper, we are going to show that this property is neither true for the Vincent-Soille's algorithm, nor for Meyer's algorithm. An important consequence of this property is that measures (such as depth (dynamics [20]), area or volume [3]) computed on the basins obtained by either Vincent-Soille's algorithm or Meyer's algorithm do not correspond to measures of connected components of lower-cross sections of the image. On the contrary, the topological watershed does allow such computations. In particular, in the case of the dynamics, a consequence of property 1 is the following result.
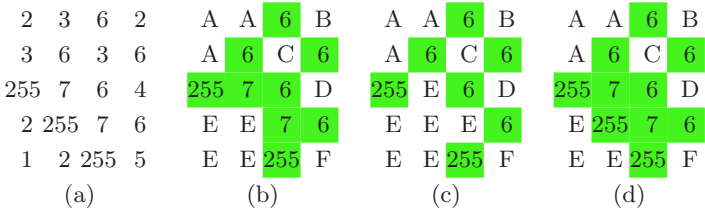
```
  2   3   6   2     A   A   6   B     A   A   6   B     A   A   6   B
  3   6   3   6     A   6   C   6     A   6   C   6     A   6   C   6
255   7   6   4    255  7   6   D    255  E   6   D    255  7   6   D
  2 255   7   6     E   E   7   6     E   E   E   6     E  255  7   6
  1   2 255   5     E   E  255  F     E   E  255  F     E   E  255  F
        (a)               (b)               (c)               (d)
```

Fig. 1. Counter-example to pass-values preservation. A greyscale image (a) and some results of watershed algorithms: (b) Vincent-Soille (c) Meyer and (d) Cost-based and Topological watershed. One can see that the pass value between E and any other basin is 6 in (c) and is 7 in (b). Both the cost-based and the topological watershed (d) do preserve the correct pass value of 255

Property 2 *Let $F \in \mathcal{F}(E)$ and let W be a topological watershed of F. Then the dynamics of a minimum for F is equal to the dynamics of the corresponding minimum for W.*

We can also prove that, for suitable cost functions, cost based watersheds of an image F preserve the pass values of F. But, as we will see, cost based watershed produces very thick contours that prevent it to be used for a saliency algorithm.

3.3   Comparison and Counter Examples for Other Watershed Algorithms

We are going to examine the behavior of watershed algorithms on several examples. In the sequel, the watershed examples are computed in 4-connectivity. In particular, regional minima are 4-connected subsets of $\mathbb{Z}^2$. On all the pictures, the basins are labeled with letters, and the watershed pixels are given with their corresponding value in the original image. Similar configurations can be found for other connectivities. Let us emphasize that configurations similar to the examples presented in this paper were found in real images.

Neither Vincent-Soille's nor Meyer's algorithm do preserve the pass values. A counter-example that illustrates this behavior is given in Figure 1. Figure 1.a presents a high contour at altitude 255. This contour is run over by the flooding principle of both Meyer and Vincent-Soille. This is especially visible on Meyer's algorithm, as in figure 1.c, the pass value between E and any other minima is 6 instead of 255. Vincent-Soille's algorithm, while having the same kind of problem, tries to detect special pixel configurations called buttonholes, and thus produces thick lines. But in this case, the Vincent-Soille's watershed is not thick enough, and the pass value between E and any other basin is 7 for the watershed contours, while it is 255 for the original image. The only correct result is produced both by the topological watershed and the cost based watershed, and is presented in figure 1.d.

Vincent-Soille's watershed algorithm aims at detecting watershed *areas* such as buttonholes. These areas are such that one cannot decide towards which
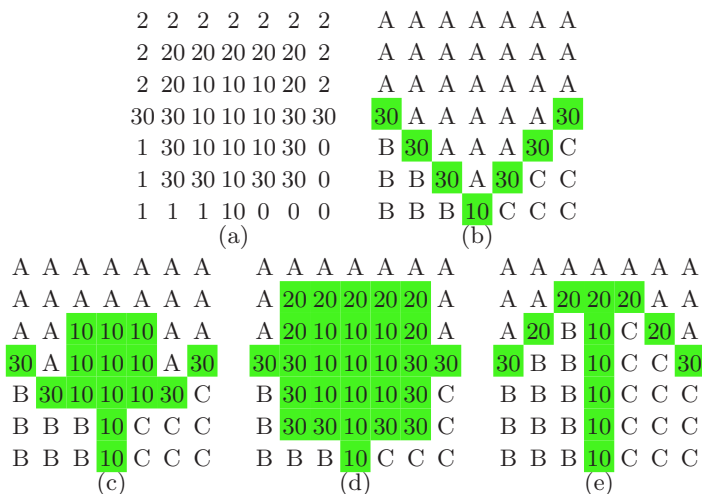
```
 2  2  2  2  2  2  2        A  A  A  A  A  A  A
 2 20 20 20 20 20  2        A  A  A  A  A  A  A
 2 20 10 10 10 20  2        A  A  A  A  A  A  A
30 30 10 10 10 30 30       30  A  A  A  A  A 30
 1 30 10 10 10 30  0        B 30  A  A  A 30  C
 1 30 30 10 30 30  0        B  B 30  A 30  C  C
 1  1  1 10  0  0  0        B  B  B 10  C  C  C
         (a)                         (b)
```

```
 A  A  A  A  A  A  A        A  A  A  A  A  A  A        A  A  A  A  A  A  A
 A  A  A  A  A  A  A        A 20 20 20 20 20  A        A  A 20 20 20  A  A
 A  A 10 10 10  A  A        A 20 10 10 10 20  A        A 20  B 10  C 20  A
30  A 10 10 10  A 30       30 30 10 10 10 30 30       30  B  B 10  C  C 30
 B 30 10 10 10 30  C        B 30 10 10 10 30  C        B  B  B 10  C  C  C
 B  B  B 10  C  C  C        B 30 30 10 30 30  C        B  B  B 10  C  C  C
 B  B  B 10  C  C  C        B  B  B 10  C  C  C        B  B  B 10  C  C  C
        (c)                        (d)                        (e)
```

Fig. 2. Another counter-example to pass values preservation. (a): Original "button-hole" image, (b): Meyer's watershed contours, (c): Vincent-Soille's watershed contours, (d): cost-based watershed contours, (e): Topological watershed contours. One can note that the contour at altitude 20 is neither kept by Vincent-Soille's algorithm, nor by Meyer's algorithm. One can also note that both the cost-based and the topological watersheds preserve the pass values of the buttonhole (a), but the topological watershed (e) is thinner than the cost-based watershed (d)

minimum a drop falling on them will slide. Figure 2.a exhibits a particular case of buttonhole. Clearly, the pixels at altitude 20 are essential since they carry the pass value between the minimum A (level 2) and the minima B and C (level 1 and 0). We can observe on figures 2.b and 2.c that both Meyer's algorithm and Vincent-Soille's remove the contour at altitude 20; in fact, Meyer's algorithm does not "see" at all this buttonhole. In both cases, pass value between A and B or C is at an altitude of 10 instead of 20 for the watershed contours.

In order to preserve pass values on the buttonhole, we have two possibilitites:

– either keeping in the watershed lines all the pixels of the buttonhole: that is what is done by the cost-based watershed (Fig. 2.d), which produces contours that cover the whole buttonhole;
– or making a careful (but arbitrary) choice between all the contours possible in the buttonhole, the choice being such that it preserves the pass values. This is what is done by the topological watershed (Fig. 2.e).

On real images, both cost based and Vincent-Soille's watershed are very sensitive to buttonholes and the resulting watershed lines can cover a large part of the image [*example not shown due to space constraint*]. Meyer's algorithm and the topological watershed compute thinner lines.

Furthermore, cost-based watersheds produce very thick lines even in the absence of buttonhole, as noted in [10], and tend to isolate basins. Figure 3 illus-
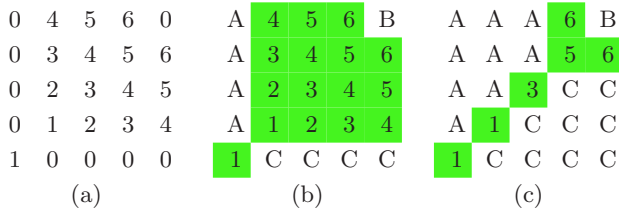
0	4	5	6	0		A	4	5	6	B		A	A	A	6	B
0	3	4	5	6		A	3	4	5	6		A	A	A	5	6
0	2	3	4	5		A	2	3	4	5		A	A	3	C	C
0	1	2	3	4		A	1	2	3	4		A	1	C	C	C
1	0	0	0	0		1	C	C	C	C		1	C	C	C	C
	(a)						(b)						(c)			

Fig. 3. A greyscale image (a) and some results of watershed algorithms:(b) cost-based watershed and (c) result according to Vincent-Soille, Meyer or Topological watershed. Basin B is isolated in (b)

trates this problem. Indeed, those algorithms have been designed to compute basins, and not lines. Thus, they cannot be used as an entry for a saliency algorithm.

Let us note that all watershed algorithms can produce thick watershed lines in some configurations (for instance, think of 4 lines crossing at one point).

4   Consequences and Conclusion

In this paper, we have shown that

– Meyer's and Vincent-Soille's algorithms do not preserve important topological features of the image; in particular, pass values are not correct. Only cost-based and topological watersheds are correct from this point of view;
– furthermore, cost-based watershed and Vincent-Soille's algorithm can produce very thick watershed lines.

Thus, only one watershed notion, the topological watershed, is suited to our task: the associated algorithm is the only one that produces a correct basin neighborhood map and correct pass values.

For computing saliency, we need a measure of contrast of the watershed basins, such as depth (dynamics [20]), area or volume [3]. An important consequence of the results of this paper is that measures computed on the basins obtained by either Vincent-Soille's algorithm or Meyer's algorithm do not correspond to measures of connected components of lower-cross sections of the image. On the contrary, we have seen that the topological watershed allows such computations. Thus, it is not possible to use the propagation mechanism of the line-building versions of Meyer's or Vincent-Soille's algorithms to compute "on-the-fly" such a measure. Such a mechanism was implemented in the Najman-Schmitt's saliency algorithm [1,2], and has also been proposed in [22], leading to non-correct results.

We could think that we can correct past saliency algorithms by replacing their watershed operator by the topological watershed. Unfortunately, this is not enough. In future papers of the series, we are going to review past saliency algorithms, to show what hypothesis they rely on are wrong, and propose a novel efficient saliency algorithm.

References

1. Najman, L.: Morphologie Mathématique: de la Segmentation d'Images à l'Analyse Multivoque. PhD thesis, Université Paris-Dauphine (1994)
2. Najman, L., Schmitt, M.: Geodesic saliency of watershed contours and hierarchical segmentation. IEEE Trans. on PAMI **18** (1996) 1163–1173
3. Vachier, C.: Extraction de caractéristiques, segmentation d'images et Morphologie Mathématique. PhD thesis, École Supérieure National des Mines de Paris (1995)
4. Breen, E., Jones, R.: Attribute openings, thinnings and granulometries. Computer Vision and Image Understanding **64** (1996) 377–389
5. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. IEEE Trans. on Image Proc. **7** (1998) 555–570
6. Meijster, A., Wilkinson, M.: A comparison of algorithms for connected set openings and closings. IEEE Trans. on PAMI **24** (2002) 484–494
7. Meyer, F.: The dynamics of minima and contours. In P. Maragos, R.S., Butt, M., eds.: ISMM 3rd. Computational Imaging and Vision, Kluwer Academic Publishers (1996) 329–336
8. Lemaréchal, C., Fjørtoft, R., Marthon, P., Cubero-Castan, E.: Comments on 'geodesic saliency of watershed contours and hierarchical segmentation'. IEEE Trans. on PAMI **20** (1998) 762–763
9. Schmitt, M.: Response to the comment "geodesic saliency of watershed contours and hierarchical segmentation". IEEE Trans. on PAMI **20** (1998) 764–767
10. Roerdink, J., Meijster, A.: The watershed transform: Definitions, algorithms and parallelization strategies. Fundamenta Informaticae **41** (2000) 187–228
11. Couprie, M., Bertrand, G.: Topological grayscale watershed transform. In: SPIE Vision Geometry V Proceedings. Volume 3168. (1997) 136–146
12. Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE Trans. on PAMI **13** (1991) 583–598
13. Meyer, F.: Un algorithme optimal de ligne de partage des eaux. In: Actes du 8ème Congrès AFCET, Lyon-Villeurbanne, France (1991) 847–859
14. Meyer, F.: Topographic distance and watershed lines. Signal Processing **38** (1994) 113–126 Special issue on Mathematical Morphology.
15. Najman, L., Schmitt, M.: Watershed of a continuous function. Signal Processing **38** (1994) 99–112 Special issue on Mathematical Morphology.
16. Lotufo, R.A., Falcao, A.X., Zampirolli, F.A.: Ift-watershed from gray-scale marker. In: SIBGRAPI'02, Fortaleza-CE, Brazil (2002) 146–152
17. Goetcherian, V.: From binary to grey tone image processing using fuzzy logic concepts. Pattern Recognition **12** (1980) 7–15
18. Bertrand, G., Everat, J., Couprie, M.: Image segmentation through operators based upon topology. Journal of Electronic Imaging **6** (1997) 395–405
19. Couprie, M., Bezerra, F.N., Bertrand, G.: Topological operators for grayscale image processing. Journal of Electronic Imaging **10** (2001) 1003–1015
20. Grimaud, M.: A new measure of contrast: Dynamics. In: SPIE Vol. 1769, Image Algebra and Morphological Processing III, San Diego (1992) 292–305
21. Najman, L., Couprie, M.: Topological watershed and contrast preservation. Discrete Applied Mathematics (2003) In preparation, special issue on DGCI 2003.
22. Andrade, M.: A topological image segmentation method by attributes and applications. PhD thesis, Universidade Federal de Minas Gerais (Brazil) (1998)