Topological Grayscale Watershed Transformation

Michel Couprie and Gilles Bertrand

Laboratoire PSI, Groupe ESIEE
Cité Descartes, BP 99
93162 Noisy-le-Grand Cedex FRANCE

Reference: SPIE Vision Geometry VI Proceedings, Vol. 3168, pp. 136-146, 1997.

Revised: october 2003.

ABSTRACT

We propose an original approach to the watershed problem, based on topology. We introduce a "one-dimensional" topology for grayscale images, and more generally for weighted graphs. This topology allows us to precisely define a topological grayscale transformation that generalizes the action of a watershed transformation. Furthermore, we propose an efficient algorithm to compute this topological grayscale transformation, and we give an example of application to image segmentation.

Keywords: discrete topology, mathematical morphology, cross-section topology, watershed, connected operator, component tree, segmentation

1. INTRODUCTION

The watershed transformation was introduced as a tool for segmenting grayscale images by S. Beucher and C. Lantuéjoul[1] in the late 70's, and is now used as a fundamental step in many powerful segmentation procedures. Efficient watershed algorithms based on immersion simulations were proposed by L. Vincent[2] and F. Meyer[3,4] in the early 90's.

Let us consider a grayscale image as a topographical relief: the gray level of a pixel becomes the elevation 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. The watershed line may be intuitively introduced as the set of points where a drop of water, falling there, may flow down towards several catchment basins of the relief.

In a recent paper[5], we introduced some basic topological notions for 2D grayscale images, which extend the classical topology of 2D binary images by considering the different *cross-sections* (thresholds) of the relief that corresponds to the image. The notion of kernel was also introduced, it corresponds to an ultimate simplification of an image which preserves its topology.

In this paper, we propose an original approach to the watershed problem, based on topological notions. We introduce a "one-dimensional" topology for grayscale images, and more generally for weighted graphs. A kernel of an image for this topology keeps the topological characteristics of a watershed transformation. A comparison with existing watershed algorithms is done. We also indicate how to build an efficient algorithm to perform this transformation, based on a tree structure that has been shown to be of general interest for image analysis[6]. Finally, we present an example of a segmentation scheme that takes advantage of this tree structure to efficiently implement both the "filtering step" and the watershed step: some powerful filters called connected operators, from the field of Mathematical Morphology, are used in order to select the "significant components" before applying the watershed transformation.

Further author information -
M.C: Email: coupriem@esiee.fr; WWW: http://www.esiee.fr/∼coupriem
G.B: Email: bertrand@esiee.fr

$$\begin{array}{ccccccc}
0 & \underline{0} & 0 & \underline{0} & 0 & \underline{0} & 0 \\
\underline{0} & 1 & \underline{0} & 1 & 0 & 1 & \underline{0} \\
\underline{0} & 1 & \underline{0} & 1 & 0 & 1 & \underline{0} \\
\underline{0} & 1 & 1 & 1 & \underline{0} & \underline{0} & 0 \\
0 & \underline{0} & \underline{0} & \underline{0} & 0 & 0 & 0
\end{array}$$

Figure 1. A set X (1) and its complement $\overline{X}$ (0). The simple points are underlined

2. TOPOLOGICAL NOTIONS FOR WEIGHTED GRAPHS

Let $G = (E, \Gamma)$ be a (symmetric) graph, where E is a set of vertices (or points), and Γ is a mapping from E into $\mathcal{P}(E)$, where $\mathcal{P}(E)$ denotes the set of all subsets of E, which associates to each point x of E, the set $\Gamma(x)$ of points adjacent to x.

For applications to digital image processing, assume that $E = \mathcal{Z}^n (n = 2, 3)$. A subset $X \subseteq E$ represents the "object", and Γ corresponds to an adjacency relation between points of E. In $\mathcal{Z}^2$, Γ may be one of the usual adjacency relations, for example the 4-adjacency or the 8-adjacency in the square grid. Note that we do not take into account the complement of X: in that sense, the topology that we will develop is "one-dimensional".

We are interested in graph transformations that preserve the number of connected components of X. We recall briefly the usual notions of path and connected component in graphs:

Definition 1: 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".
A subset X of E is *connected* if it is made of exactly one connected component.

We will now introduce the notion of "simple point" in a graph. Intuitively, a point of $\overline{X}$ is "simple" if it may be added to X while preserving the number of connected components of X ($\overline{X}$ denotes the complement of X). This corresponds to the preservation of a "one-dimensional" topology. It is important to note that the usual definition of a simple point in $\mathcal{Z}^2$ corresponds to the preservation of a "two-dimensional" topology[7], that is to say, not only the number of connected components of X is preserved, but also the number of connected components of $\overline{X}$.

Definition 2: Let $G = (E, \Gamma)$ be a graph, and let $X \subset E$.
The point $x \in \overline{X}$ is *simple* (for X) if the number of connected components of $X \cup \{x\}$ equals the number of connected components of X.
In other words, x is simple (for X) if x is adjacent to exactly one connected component of X.

In Fig. 1, the points of the set X are represented by "1"s, and the 4-adjacency is assumed. In $\overline{X}$, simple points are underlined. It may be easily seen that one cannot locally decide whether a point is simple or not.

We will now extend this notion to a weighted graph (E, Γ, F), where F is a function from E to $\mathcal{Z}$, in order to use it for digital grayscale images. For any point $x \in E$, $F(x)$ represents the gray level of x.

Definition 3: We denote $\mathcal{F}(E)$ the set composed of all functions from E to $\mathcal{Z}$. Let $F \in \mathcal{F}(E)$.
We denote $F_k = \{x \in E; F(x) \geq k\}$ with $k \in \mathcal{Z}$; F_k is called a *(cross-)section* of F.
A connected component of a section F_k is called a *(level k) component* of F.
A component of F that has only strictly lower neighbors is called a *(regional) maximum* of F.

Fig. 2 shows a grayscale image F and three cross-sections of F. If we use the 4-adjacency, F_3 is made of two components, whereas F_2 and F_4 are made of one component. The set F_4 is a regional maximum of F.

$$
\begin{array}{ccccccc}
2&2&2&2&2&2&2\\
2&4&4&4&4&4&2\\
2&4&4&2&4&4&2\\
2&3&2&3&1&4&2\\
2&4&4&1&4&4&2\\
2&4&4&4&4&4&2\\
2&2&2&2&2&2&2\\
\multicolumn{7}{c}{F}
\end{array}
\quad
\begin{array}{ccccccc}
1&1&1&1&1&1&1\\
1&1&1&1&1&1&1\\
1&1&1&1&1&1&1\\
1&1&1&1&0&1&1\\
1&1&1&0&1&1&1\\
1&1&1&1&1&1&1\\
1&1&1&1&1&1&1\\
\multicolumn{7}{c}{F_2}
\end{array}
\quad
\begin{array}{ccccccc}
0&0&0&0&0&0&0\\
0&1&1&1&1&1&0\\
0&1&1&0&1&1&0\\
0&1&0&1&0&1&0\\
0&1&1&0&1&1&0\\
0&1&1&1&1&1&0\\
0&0&0&0&0&0&0\\
\multicolumn{7}{c}{F_3}
\end{array}
\quad
\begin{array}{ccccccc}
0&0&0&0&0&0&0\\
0&1&1&1&1&1&0\\
0&1&1&0&1&1&0\\
0&0&0&0&0&1&0\\
0&1&1&0&1&1&0\\
0&1&1&1&1&1&0\\
0&0&0&0&0&0&0\\
\multicolumn{7}{c}{F_4}
\end{array}
$$

Figure 2. A grayscale image F and some of its cross-sections

$$
\begin{array}{ccccccc}
20&20&20&20&20&20&20\\
20&20&20&5&20&20&20\\
20&20&8&13&6&20&20\\
20&9&14&16&13&7&20\\
20&9&13&18&17&9&20\\
20&9&15&17&15&9&20\\
20&20&9&15&9&20&20\\
20&20&13&9&12&20&20\\
20&20&14&9&13&20&20\\
20&20&15&9&12&20&20\\
20&20&13&9&12&20&20\\
20&20&9&15&9&20&20\\
20&9&15&17&15&9&20\\
20&9&13&19&17&9&20\\
20&9&14&16&13&7&20\\
20&20&8&13&6&20&20\\
20&20&20&5&20&20&20\\
20&20&20&20&20&20&20\\
\multicolumn{7}{c}{\text{(a)}}
\end{array}
\quad
\begin{array}{ccccccc}
20&20&20&20&20&20&20\\
20&20&20&9&20&20&20\\
20&20&9&18&9&20&20\\
20&9&18&18&18&9&20\\
20&9&18&18&18&9&20\\
20&9&18&18&18&9&20\\
20&20&9&18&9&20&20\\
20&20&20&9&20&20&20\\
20&20&20&20&20&20&20\\
20&20&20&20&20&20&20\\
20&20&20&9&20&20&20\\
20&20&9&19&9&20&20\\
20&9&19&19&19&9&20\\
20&9&19&19&19&9&20\\
20&9&19&19&19&9&20\\
20&20&9&19&9&20&20\\
20&20&20&9&20&20&20\\
20&20&20&20&20&20&20\\
\multicolumn{7}{c}{\text{(b)}}
\end{array}
\quad
\begin{array}{ccccccc}
0&0&0&0&0&0&0\\
0&0&0&15&0&0&0\\
0&0&12&7&14&0&0\\
0&11&6&4&7&13&0\\
0&11&7&2&3&11&0\\
0&11&5&3&5&11&0\\
0&0&11&5&11&0&0\\
0&0&7&11&8&0&0\\
0&0&6&11&7&0&0\\
0&0&5&11&8&0&0\\
0&0&7&11&8&0&0\\
0&0&11&5&11&0&0\\
0&11&5&3&5&11&0\\
0&11&7&1&3&11&0\\
0&11&6&4&7&13&0\\
0&0&12&7&14&0&0\\
0&0&0&15&0&0&0\\
0&0&0&0&0&0&0\\
\multicolumn{7}{c}{\text{(c)}}
\end{array}
\quad
\begin{array}{ccccccc}
0&0&0&0&0&0&0\\
0&0&0&11&0&0&0\\
0&0&11&2&11&0&0\\
0&11&2&2&2&11&0\\
0&11&2&2&2&11&0\\
0&11&2&2&2&11&0\\
0&0&11&2&11&0&0\\
0&0&0&11&0&0&0\\
0&0&0&0&0&0&0\\
0&0&0&0&0&0&0\\
0&0&0&11&0&0&0\\
0&0&11&1&11&0&0\\
0&11&1&1&1&11&0\\
0&11&1&1&1&11&0\\
0&11&1&1&1&11&0\\
0&0&11&1&11&0&0\\
0&0&0&11&0&0&0\\
0&0&0&0&0&0&0\\
\multicolumn{7}{c}{\text{(d)}}
\end{array}
$$

Figure 3. (a): original image, (b): upper-homotopic kernel of (a), (c): inverse of (a), (d): inverse of (b)

Definition 4: Let $F \in \mathcal{F}(E)$, $x \in E$, and $k = F(x)$.
The point x is *constructible* (for F) if x is simple for F_{k+1}.
In other words, each neighbor y of x, such that $F(y) > k$, belongs to the same level $k+1$ component of F.
Furthermore, if we consider $F' \in \mathcal{F}(E)$ such that $\forall y \neq x; F'(y) = F(y)$ and $F'(x) = F(x) + 1$, it may be easily seen that $\forall k \in \mathcal{Z}$, the number of connected components of F'_k equals the number of connected components of F_k. That is to say, the value of a constructible point may be raised by one without changing the number of connected components of each cross-section of F.

Definition 5: Let $F \in \mathcal{F}(E)$. We say that $G \in \mathcal{F}(E)$ is *upper-homotopic* to F if G may be derived from F by the iterative raising of constructible points.
We say that $K \in \mathcal{F}(E)$ is an *upper-homotopic kernel* of F if K is upper-homotopic to F and K is maximal for this property, that is, all points of E are non-constructible for K.

In Fig. 3, we present an image 3a and its upper-homotopic kernel 3b. Note that in 3b, the maxima 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 and 3d are inverse images (with $\forall x \in E, \overline{F}(x) = 20 - F(x)$) of 3a and 3b, respectively. We can see that the set of points that are not in a minimum of 3d correspond to the watershed line of 3c. A deeper comparison between the upper-homotopic kernel and the watershed transformation follows in section 5.

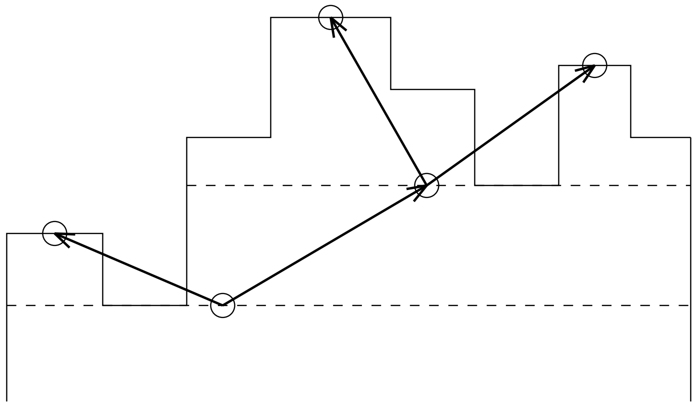

Figure 4. 1-D function and its component tree

3. COMPONENT TREE

In this section, we define the component tree structure that will allow us to propose an efficient algorithm for the upper-homotopic kernel transformation.

The use of this tree in order to represent the "meaningful" information contained in a numerical function is not new. In particular, P. Hanusse and P. Guillataud[6] claim that this tree can play a central role in image segmentation, and propose an efficient way to compute it, based on an immersion simulation. C. Vachier[8] suggests that it should be used to efficiently implement some morphological operators called "extinction functions".

Our main goal in this paper is to define an upper-homotopic kernel operator, and to propose an efficient algorithm for this operator. Basically, this algorithm will have to repeat the following actions until stability: a) select a constructible point, b) raise its value. By the very definition of a constructible point, we may raise the value of a constructible point by one. In fact, in many cases, it is possible to raise the value of a constructible point by more than one, without changing the number of level k components of F, $\forall k \in \mathcal{Z}$. It should be pointed out that checking whether a point is constructible or not cannot be done locally if the only available information is the mesh (E, Γ) and the function F. The component tree will allow to answer this question in constant time.

First, we establish a classification of the components of a function:

Definition 6: Let $F \in \mathcal{F}(E)$, a level k component of F may include either zero, one, or striclty more than one level $k+1$ components. We call:
• a *leaf* component, a component that includes no level $k+1$ component (that is, a regional maximum);
• a *branch* component, a component that includes exactly one level $k+1$ component;
• a *node* component, a component that includes strictly more than one level $k+1$ components.

We remark that the branch components are not essential to characterize the topology of F. Thus, we discard them and keep only the node and leaf components, which represent the topological discontinuities of the function F. These non-branch components, thanks to the inclusion relation, may be organized in a tree structure (see Fig. 4):

Definition 7: Let $F \in \mathcal{F}(E)$, we denote $\mathcal{C}(F)$ the set of all non-branch components of F. We define the *component tree* $\mathcal{T}(F)$ as a directed tree such that:
• $\mathcal{C}(F)$ is the set of vertices of the component tree;
• there is an edge from $C_1 \in \mathcal{C}(F)$ to $C_2 \in \mathcal{C}(F)$ if $C_2 \subset C_1$ and for each component c of F such that $C_2 \subset c \subset C_1$, c is a branch component (where $\subset$ denotes the strict inclusion relation). We denote $C_1 = \text{father}(C_2)$.

The next definition introduces the component mapping which associates, to each point of E, a component in $\mathcal{C}(F)$.

Let $F \in \mathcal{F}(E)$, $x \in E$, $k = F(x)$, we denote c_x the level k component of F (either branch or non-branch) that contains x.

Definition 8: Let $F \in \mathcal{F}(E)$, to each non-branch component $c \in \mathcal{C}(F)$, we associate a subset $I(c)$ of E called the *influence area* of c (see Fig. 5), and defined as $I(c) = \{x \in E, c \subseteq c_x \subset \text{father}(c)\}$.

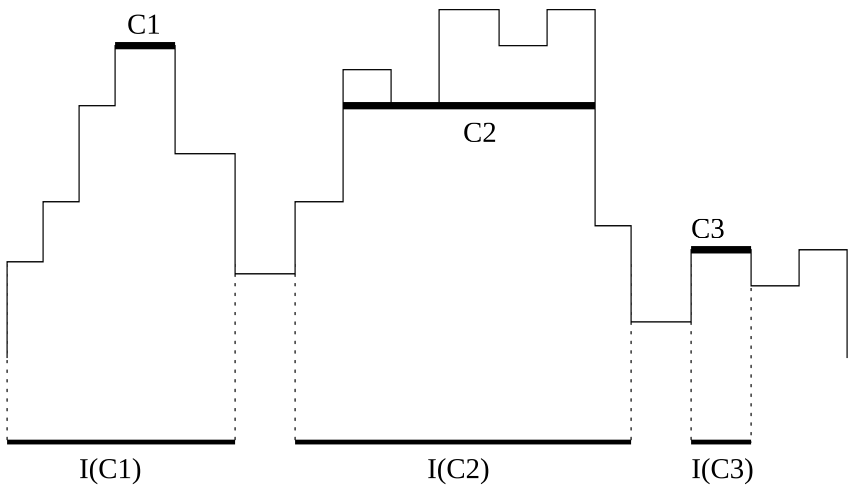

Figure 5. Some components and their influence area

Conversely, to each point x of E, we associate the highest level non-branch component $M(x)$ such that $x \in I(M(x))$. The mapping M, from E to $\mathcal{C}(F)$, is called the *component mapping* of F.

Note that $I(c)$ may be defined equivalently as the level $f+1$ component of F that includes c, f being the level of father(c).

The couple: (tree $\mathcal{T}$, mapping M) allows us to test whether a point belongs to a given component or not, since a point x belongs to a component c if and only if there exists a path from c to $M(x)$ in $\mathcal{T}$, in other words, if c is an *ancestor* of $M(x)$ in $\mathcal{T}$.

Furthermore, the next two properties give us an efficient way to characterize constructible points from the component tree $\mathcal{T}$ and the component mapping M.

Property 1: Let $F \in \mathcal{F}(E)$, let $x, y \in E$. The nearest common ancestor of $M(x)$ and $M(y)$ in the component tree $\mathcal{T}(F)$ is the highest level non-branch component that contains both $M(x)$ and $M(y)$.
Furthermore, the points x and y belong to the influence zone of the same level k component if and only if the nearest common ancestor of $M(x)$ and $M(y)$ in the component tree $\mathcal{T}(F)$ has a level $k' \geq k$.

Property 1 may be easily generalized to any number of points of E.

Property 2: Let $F \in \mathcal{F}(E)$, $x \in E$, $k = F(x)$, and let $\Gamma^+(x) = \{y \in \Gamma(x), F(y) > F(x)\}$, the set of neighbors of x "strictly higher" than x. We denote $\mathcal{M}(x) = \{M(y), y \in \Gamma^+(x)\}$.
The point x is constructible (for F) if and only if the nearest common ancestor of all the components in $\mathcal{M}(x)$ in the component tree $\mathcal{T}(F)$ has a level $k' \geq k+1$.

The problem of finding the nearest common ancestor of a set of nodes in a directed tree has been well studied, and efficient algorithms exist: D. Harel and R.E. Tarjan[9] showed that one can build in linear time a representation of a tree, which allows to find the nearest common ancestors of a set of nodes in constant time.

4. ALGORITHM FOR AN UPPER-HOMOTOPIC KERNEL

Our algorithm to get an upper-homotopic kernel from an image consists in two steps:

Upper-homotopic kernel algorithm:
1) construct the component tree $\mathcal{T}$ and the mapping M from the original image F.
2) iteratively select a constructible point and raise it, thanks to the informations in ($\mathcal{T}$, M) and using Property 2, then update the mapping M. This step must be repeated until stability is reached.

By using an "immersion-like" strategy, and a hierarchical queue structure[6,3,4], the complexity of the construction of the component tree (step 1) may be reduced to the complexity of the union-find algorithm that was analysed by R.E. Tarjan[10]. This complexity is almost-linear. Step 2 may be efficiently implemented by using a "breadth-first" strategy.

A *topological grayscale watershed* of an image F may be obtained by taking the inverse of an upper-homotopic kernel computed from the inverse of F (see Fig. 3).

An important feature of the watershed algorithms is to allow the use of a set of "markers" that identify the "significant basins" of the relief. When using such markers, the minima and basins that are not marked may be seen as "filled in".

In our approach, some components of the component tree may also be marked, and other ones eliminated, before performing the step 2 of the upper-homotopic kernel algorithm. Furthermore, the component tree may be used to efficiently implement connected operators, which are powerful tools that allow to find out markers. This last point is developped in Section 6.

5. COMPARISON WITH MEYER'S WATERSHED ALGORITHM

The watershed line is often depicted in an intuitive manner as the set of points such that a drop of water, falling there, may flow down to several different minima of the relief.

Despite its name, the watershed line is not always a "thin" set. Consider a plateau between two basins: every point of the plateau is a watershed point, according to the latter point of view. Such non-thin watershed sets are undesirable for the purpose of segmenting images, this is why all the proposed watershed algorithms manage to get a thin watershed line in a plateau. In order to precisely define the location of the watershed line in such cases, a distance criterion has been introduced[11,12]. It should be noted that such a distance criterion may be also easily used in our algorithm.

Nevertheless, this last enhancement does not eliminate all thick parts of the watershed set. Some configurations may occur, either in the continuous case or in the discrete case, where one cannot decide towards which basin a drop of water will flow down. Fig. 6a shows such a configuration, and similar ones may be encountered in some real images. L. Vincent[2] claims that a non-thin watershed set is the correct answer for such configurations. Nevertheless, classical watershed algorithms such as Meyer's[3] manage to get a thin set even in such cases. We recall the principle of Meyer's algorithm:

Meyer's watershed algorithm:
Starting from a grayscale image F and a set of markers with different labels,
1) insert every neighbor x of every marked area in a hierarchical queue, with a priority level corresponding to the gray 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 gray level. If $\Gamma(x)$ 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 result W is the complement of the set of labeled points.

Let M be the set of the minima of the original image F. Meyer's algorithm expands as much as possible the set M, while preserving the number of connected components of M. Thus, W is a kernel of M for the one-dimensional topology we introduced on graphs. Furthermore, thanks to the immersion strategy, the points of W are located in such a way that they have the highest possible value in the image F.

It is important to note that the result W of this algorithm is a set, whereas a topological grayscale watershed is a function. In order to compare the effects of both operators, let us consider the set S of points that do not belong to a minimum of a topological grayscale watershed.

It may easily be seen that the sets W and S are equivalent up to a topological transformation. Furthermore, in most cases, the sets W and S are very close to each other, as illustrated in Fig. 7.

Nevertheless, in some configurations, there is a significative difference between W and S. An example of such a configuration is depicted Fig. 6a. In this configuration, a plateau P is composed of points at the level 10. Intuitively,

30	30	30	30	30	30	30	30	30
30	2	2	2	2	2	2	2	30
30	2	20	20	20	20	20	2	30
30	2	20	10	10	10	20	2	30
30	30	30	10	10	10	30	30	30
30	1	30	10	10	10	30	0	30
30	1	30	30	10	30	30	0	30
30	1	1	1	10	0	0	0	30
30	30	30	30	30	30	30	30	30

(a)

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

(b)

30	30	30	30	30	30	30	30	30
30	2	2	2	2	2	2	2	30
30	2	2	20	20	20	2	2	30
30	2	20	1	1	1	20	2	30
30	20	1	1	1	1	1	20	30
30	1	1	1	1	1	10	0	30
30	1	1	1	1	10	0	0	30
30	1	1	1	10	0	0	0	30
30	30	30	30	30	30	30	30	30

(c)

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

(d)

Figure 6. (a): original image, (b): Meyer's watershed line, (c): topological grayscale watershed, (d): non-minima of the topological grayscale watershed

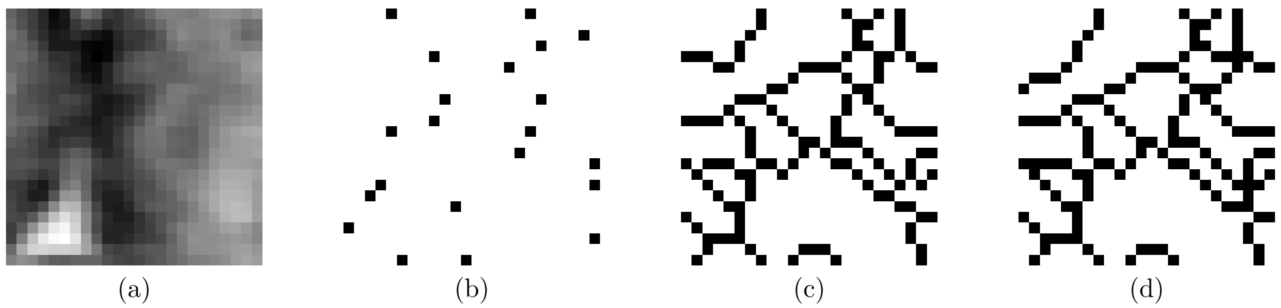

Figure 7. (a): original image, (b): minima of (a), (c): Meyer's watershed line, (d): non-minima of the topological grayscale watershed

a drop of water falling inside P may flow down towards the minimum 0 or towards the minimum 1, but cannot flow down towards the minimum 2. Therefore, the watershed line in 6d is better placed than the one in 6c: Meyer's algorithm "assigns" the points of P to the minimum 2, while the topological watershed transform "assigns" the points of P either to the minimum 1 or 0.

6. APPLICATION TO SEGMENTATION

We will now present a segmentation scheme based on connected operators for the filtering step, and on the topological grayscale watershed transformation for the reconstruction step. Connected operators are powerful tools for image segmentation that were developped in the framework of the Mathematical Morphology[13,14,15]. They allow to filter regions according to some criteria such as area or contrast. We recall here briefly the minimal set of definitions needed to introduce this notion:

Definition 9: Let E be a set, a transformation ψ acting on $\mathcal{P}(E)$ is:
- *increasing* if $X \subseteq Y \Rightarrow \psi(X) \subseteq \psi(Y), \forall X, Y \subseteq E$;
- *idempotent* if $\psi(\psi(X)) = \psi(X), \forall X \subseteq E$;
- *anti-extensive* if $\psi(X) \subseteq X, \forall X \subseteq E$.

ψ is an *opening* if it is increasing, idempotent and anti-extensive.

Definition 10: Let (E, Γ) be a graph, $x \in E$, we define the *connected opening* ψ_x as follows:

$$\forall X \subseteq E, \psi_x(X) = \begin{cases} \text{the connected component of } X \text{ that contains } x, & \text{if } x \in X \\ \emptyset, & \text{otherwise} \end{cases}$$

Definition 11: Let E be a set. A *criterion* on E is a mapping from $\mathcal{P}(E)$ to {false, true}.
A criterion T on E is *increasing* if:

$$\forall X, Y \subseteq E, \begin{cases} T(X) = \text{true and } Y \supseteq X & \Rightarrow T(Y) = \text{true} \\ T(X) = \text{false and } Y \subseteq X & \Rightarrow T(Y) = \text{false} \end{cases}$$

For example, $T(X) \equiv [\text{area}(X) \geq \lambda]$ is an increasing criterion.

Definition 12: Let E be a set, let T be an increasing criterion on E. We define the *trivial opening* ψ_T as follows:

$$\forall X \subseteq E, \psi_T(X) = \begin{cases} X & \text{if } T(X) = \text{true} \\ \emptyset & \text{otherwise} \end{cases}$$

Definition 13: Let (E, Γ) be a graph, let T be an increasing criterion on E.
We define the *attribute opening* ψ^T as follows:

$$\forall X \subseteq E, \psi^T(X) = \bigcup_{x \in X} \psi_T(\psi_x(X))$$

The attribute opening ψ^T "filters" the connected regions that satisfy the criterion T.

Definition 14: Let (E, Γ) be a graph, let $F \in \mathcal{F}(E)$, let ψ^T be an attribute opening on E.
We define Ψ^T, which is called a *connected operator*, as follows:

$$\forall x \in E, \Psi^T(F)(x) = \max\{k \in \mathcal{Z}, x \in \psi^T(F_k)\}$$

In the framework of functions, some basic increasing criteria may be considered:

Definition 15: Let (E, Γ) be a graph, let $F \in \mathcal{F}(E)$, let $k, \lambda \in \mathcal{Z}$.

$\forall X \subseteq F_k,$	$\text{area}(X) = \text{card}(X),$	$A(X) \equiv [\text{area}(X) \geq \lambda]$
$\forall X \subseteq F_k,$	$\text{height}(X) = \max\{F(x) - k, x \in X\},$	$H(X) \equiv [\text{height}(X) \geq \lambda]$
$\forall X \subseteq F_k,$	$\text{volume}(X) = \sum_{x \in X}(F(x) - k),$	$V(X) \equiv [\text{volume}(X) \geq \lambda]$

A typical segmentation scheme based on connected operators consists in the following steps:
(1) to an image F, apply a connected operator $\Psi^T(F)$ according to an increasing criterion T,
(2) extract the maxima from $\Psi^T(F)$,
(3) reconstruct the geometry of the segmented regions, by using a watershed operator with the extracted maxima as marker set.

We will now see how to implement efficiently a topological equivalent of steps (1) and (2) with the help of the component tree. First, we must associate, to each component C in the tree, a "measure" of this component. In fact, this measure does not correspond to the component C itself, but rather to its influence area $I(C)$, for example:

$$\mu(C) = \text{area}(I(C)), \qquad \text{or} \qquad \mu(C) = \text{height}(I(C)), \qquad \text{or} \qquad \mu(C) = \text{volume}(I(C))$$

These informations may be computed and stored during the construction of the component tree. Then, the marking algorithm is as follows:

Marking algorithm:
The leaves of the component tree (regional maxima) are put into a list l_1. Another list l_2 is set to empty.
1) A component C is drawn from l_1. If $\mu(C) \geq \lambda$ and if C has not been invalidated, then C is marked as "pertinent" and the ancestors of C are invalidated. If $\mu(C) < \lambda$, then father(C) is put into l_2.
Step 1 must be repeated until l_1 is empty.
2) If l_2 is not empty, swap l_1 and l_2, and go to step 1.

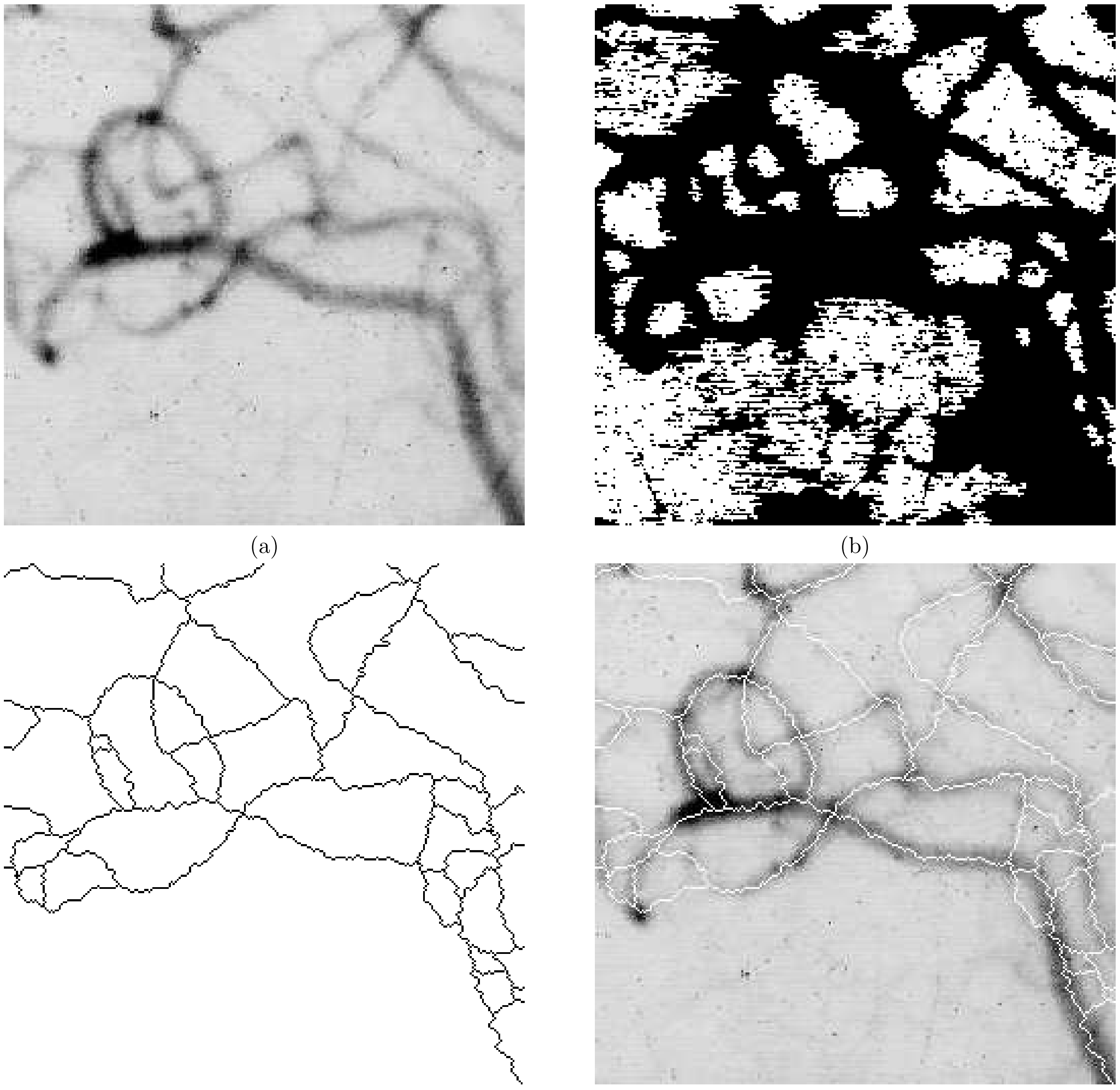

Figure 8. (a): original image, (b): marked components, (c): maxima of the upper-homotopic kernel (with a marking step), (d): superposition with the original image

7. RESULTS

Figure 8 shows the results of this segmentation scheme on a blood vessel image (angiography). First, the component tree of the original is built. Then, some components are selected by using the marking algorithm. Here, two criteria are used: heigth $\geq$ 15 and area $\geq$ 10. The selected components appear in white Fig. 8b. The last step is the upper-homotopic kernel. Fig. 8c shows the maxima of the result, and Fig. 8d is a superposition of 8c (inverted) and 8a.

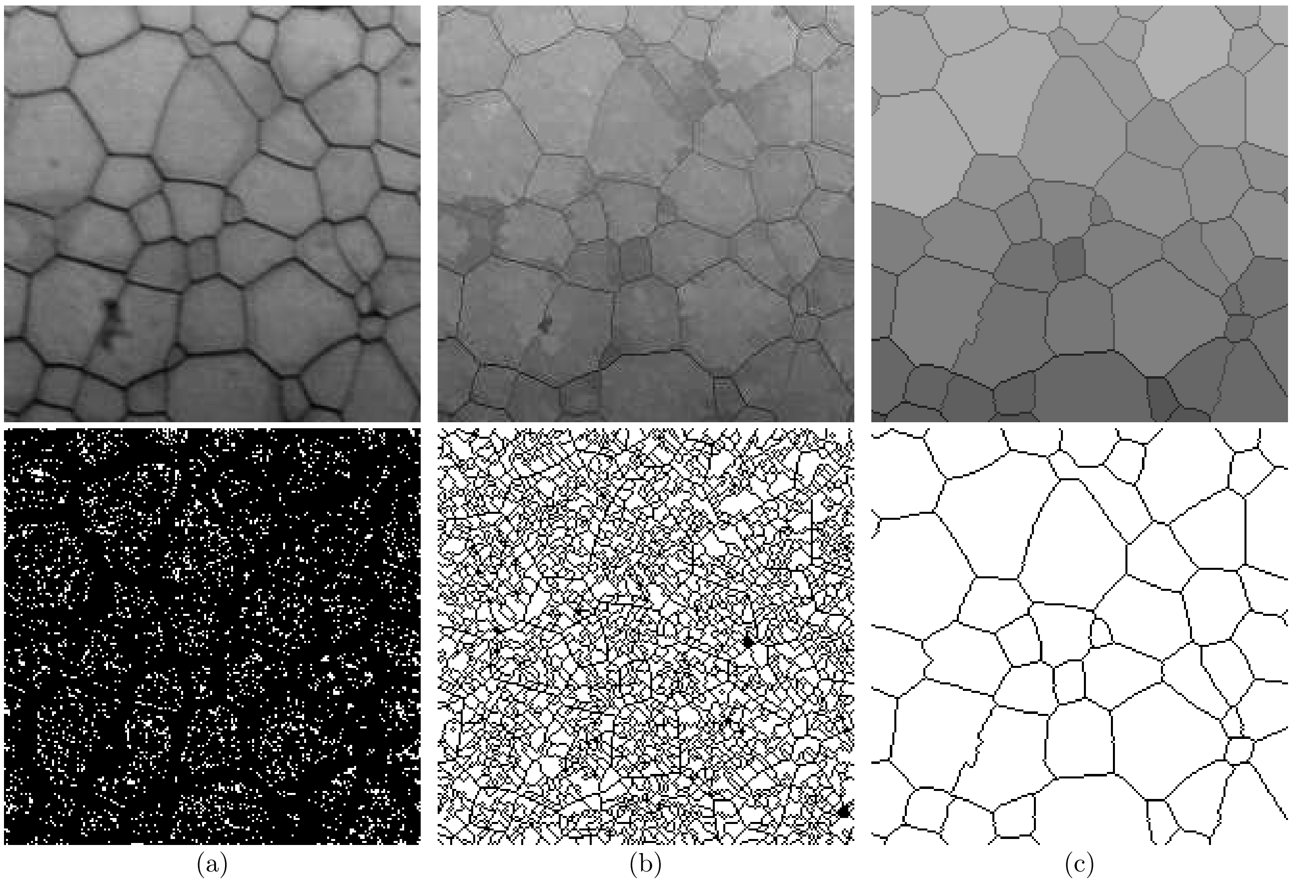

Figure 9. (a): original image, (b): upper-homotopic kernel, (c): upper-homotopic kernel with a marking step (height ≥ 23) — below: the maxima

In Fig. 9, the effects of the upper-homotopic kernel, without (9b) and with (9c) a marking step, are illustrated with an uranium oxyde micrography. Below each grayscale image, the corresponding regional maxima are given: they appear in white. We can see that, in a real image, there is a lot of maxima, and each of these regions is composed of only few points. The upper-homotopic kernel expands as much as possible these maxima (9b). Using the marking algorithm allow to filter the "significant" maxima (9c).

It should be noted that the upper-homotopic kernel 9b is a grayscale image, and, though still strongly over-segmented, it contains a lot of pertinent informations that may be used for further treatments. In particular, its component tree is the same as the original image's component tree. Furthermore, in an upper-homotopic kernel, the maxima come into contact through "thin" frontier lines. This fact may be exploited by some *regularization operators*, that can eliminate the "non-significant" maxima or frontier lines without the need of defining or tuning any parameter. We developped such an approach in recent papers[5,16,17].

8. CONCLUSION

In this paper, we have introduced a topological grayscale watershed transformation that generalizes the existing watershed transformation. This new operator acts on a weighted graph, and is based on the definition of a "one-dimensional" topology for graphs and weighted graphs. When applied to a grayscale image, its output is also a grayscale image that still contains pertinent topological and grayscale information. It may be easily combined with a distance criterion in order to precisely define the position of the watershed line in plateaus.

Furthermore, the topological grayscale watershed transformation may be efficiently computed thanks to the component tree, which is a very useful representation of the topology of a weighted graph. The time complexity of the construction of the component tree is almost-linear. We showed in this paper that, based on this tree, a complete segmentation scheme using a connected operator for filtering and using the topological grayscale watershed transformation for reconstruction may be efficiently implemented.

Aknowledgments We thank Jean Cousty for his careful reading of the paper, which led to the correction of some errors.

REFERENCES

1. S. Beucher, Ch. Lantuejoul, "Use of Watersheds in Contour Detection", *Proc. Int. Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation*, Rennes, France, 1979.
2. L. Vincent, P. Soille, "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 6, pp. 583-598, 1991.
3. F. Meyer, "Un algorithme optimal de ligne de partage des eaux", *8th Conf. Reconnaissance des Formes et Intelligence Artificielle*, Vol. 2, pp. 847-859, AFCET Ed., Lyon, 1992.
4. S. Beucher and 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.
5. G. Bertrand, J.C. Everat and M. Couprie, "A Topological approach to image segmentation", *SPIE, Vision Geometry V*, Vol. 2826, pp. 65-76, 1996.
6. P. Hanusse, P. Guillataud, "Sémantique des images par analyse dendronique", *8th Conf. Reconnaissance des Formes et Intelligence Artificielle*, Vol. 2, pp. 577-588, AFCET Ed., Lyon, 1992.
7. T.Y Kong and A. Rosenfeld, "Digital topology: introduction and survey", *Computer Vision, Graphics and Image Processing*, 48, pp. 357-393, 1989.
8. C. Vachier, "Extraction de caractéristiques, segmentation d'images et Morphologie Mathématique", PhD Thesis, École des Mines, Paris, 1995.
9. D. Harel, R.E. Tarjan, "Fast Algorithms for Finding Nearest Common Ancestors", *SIAM J. Comput.*, Vol. 13, No. 2, pp. 338-355, 1984.
10. R.E. Tarjan, *Data Structures and Network Algorithms*, SIAM, 1978.
11. F. Preteux, "Watershed and Skeleton by Influence Zone: A Distance-Based Approach", *Journal of Mathematical Imaging and Vision*, No. 1, pp. 239-255, 1992.
12. F. Meyer, "Topographic distance and watershed lines", *Signal Processing*, No. 38, pp. 113-125, 1994.
13. J. Serra, *Image Analysis and Mathematical Morphology, Vol. II: Theoretical Advances*, Academic Press, 1988.
14. J. Serra, L. Vincent, "An Overview of Morphological Filtering", *Circuits Systems Signal Process*, Vol. 11, No. 1, pp. 48-107, 1992.
15. E.J. Breen, R. Jones, "Attribute Openings, Thinnings and Granulometries", *Computer Vision and Image Understanding*, Vol. 64, No. 3, pp. 377-389, 1996.
16. J.C. Everat and G. Bertrand, "New topological operators for segmentation", *Proc. ICIP'96*, Vol. 3, pp. 45-48, IEEE Signal Processing Society, 1996.
17. J.C. Everat, G. Bertrand, and M. Couprie, "Reconstruction operators for image segmentation", to appear in *Proc. 3rd Int. Workshop on Visual Form*, IAPR, Capri 1997.