Chapter 12

The Morphological Approach to Segmentation: The Watershed Transformation

S. Beucher and F. Meyer

Centre de Morphologie Mathématique
Ecole des Mines de Paris
Fontainebleau, France

I. INTRODUCTION

Segmentation is one of the key problems in image processing. In fact, one should say segmentations because there exist as many techniques as there are specific situations. Among them, gray-tone images segmentation is very important and the relative techniques may be divided into two groups: the techniques based on contour detection and those involving region growing. Many authors have tried to define general schemes of contour detection using low-level tools [1,2]. Unfortunately, because they work at a very primitive level, a great number of algorithms must be used to emphasize their results.

An original method of segmentation based on the use of watershed lines has been developed in the framework of mathematical morphology. This technique, which may appear to be close to the region-growing methods, leads in fact to a general methodology of segmentation and has been applied with success in many different situations.

In this chapter, the principles of morphological segmentation will be presented and illustrated by means of examples, starting from the simplest ones and introducing step by step more complex segmentation tools.

In Section II, we shall review briefly various morphological tools which are used throughout this chapter. These basic transformations are useful for the description of some algorithms used in morphological segmentation. We shall not introduce the basic notions of mathematical morphology; the reader not familiar with them is invited to refer to [3,4].

Section III will be devoted to the presentation of the watershed lines and to their use in segmentation through a very simple didactic example. A simple watershed algorithm will be described.

A real segmentation problem will be presented in Section IV. The problems which arise will be discussed and solved by means of the second great morphological tool used in segmentation: homotopy modification.

In Section V, some algorithms for watershed construction and for homotopy modification will be described. However, the computational cost is the major drawback of the method. Hence the optimality and speed of the algorithms become a critical issue.

In Section VI, various examples of segmentations taken in many domains of image analysis will be discussed.

At this point, a general scheme for segmentation using mathematical morphology will be introduced. More complex algorithms based on a hierarchical approach to the segmentation will be presented. Then examples of complex segmentation will be given.

Finally, the advantages and drawbacks of this methodology will be discussed.

Although we will try in this chapter to be as complete as possible, it is not possible to give an extensive presentation of all the existing techniques of morphological segmentation. Such a review may be found in [5] or in an introductory paper by the authors [6].

II. A REVIEW OF SOME BASIC TOOLS

A. Notation

For simplicity, we will mainly present the segmentation tools in the framework of digital pictures. In this representation, a graytone image can be represented by a function $f : \mathbf{Z}^2 \rightarrow \mathbf{Z}$. $f(x)$ is the gray value of the image at point x. The points of the space $\mathbf{Z}^2$ may be the vertices of a square or of a hexagonal grid.

A section of f at level i is a set $X_i(f)$ defined as

$$X_i(f) = \{x \in \mathbf{Z}^2 : f(x) \geq i\}$$

In the same way, we may define the set $Z_i(f)$:

$$Z_i(f) = \{x \in \mathbf{Z}^2 : f(x) \leq i\}$$

We have obviously

$$X_i(f) = Z^c_{i+1}(f)$$

We shall denote by $X \oplus B$ (resp. $f \oplus B$) the dilation of a set X (resp. a function f) by an elementary disk B (square or hexagon) and by $X \ominus B$ (resp. $f \ominus B$) the elementary erosion. The corresponding opening and closing by the same

structuring element are denoted respectively by X_B and X^B. We shall also denote by γ and φ some general morphological openings and closings.

B. Definition of Some Basic Transformations

In this section, some useful morphological tools for segmentation are described: gradient, top-hat transform, distance function, geodesic distance function, and more generally the geodesic reconstructions. Then the notion of homotopy and homotopic transformations are introduced.

The gradient image (or the top-hat transform) is often used in the watershed transformation, because the main criterion for the segmentation in many applications is the homogeneity of the gray values of the objects present in the image. But other criteria may be relevant and other functions may be used. In particular, when the segmentation is based on the shape of the objects, the distance function is very helpful.

The geodesic transformations are of primary importance for the explanation of both the watershed and the homotopy modification algorithms. Among these transformations, the geodesic skeleton by zones of influence and the reconstruction (both for sets and for functions) are fundamental approaches.

1. Morphological Gradient

The morphological gradient [5] of a picture is defined as

$$g(f) = (f \oplus B) - (f \ominus B)$$

When f is continuously differentiable, this gradient is equal to the modulus of the gradient of f (Figure 1):

$$g(f) = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}$$

The simplest way to approximate this modulus is to assign to each point x the difference between the highest and the lowest pixels within a given neighborhood of x. In other words, for a function f, it is the difference between the dilated function $f \oplus B$ and the eroded function $f \ominus B$.

2. The Top-Hat Transformation

The top-hat transform WTH(f) of a function f is defined as the difference between the function and its morphological opening [7]:

$$\text{WTH}(f) = f - \gamma(f)$$

This transformation is a very good contrast detector suitable for enhancing the white and narrow objects in the image (Figure 2). Different sizes and shapes may be chosen for the structuring element used in the opening and this leads to very

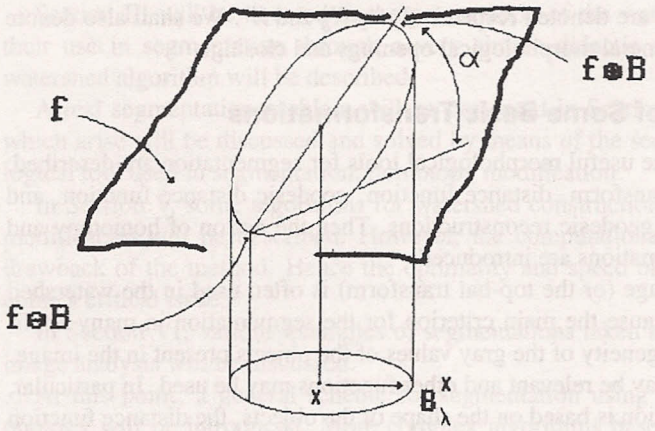
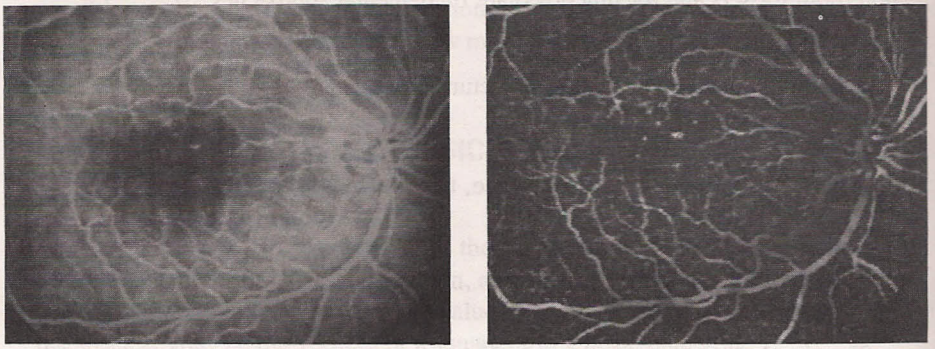


Figure 1. Construction of the morphological gradient.



(a) (b)

Figure 2. White top-hat transform (b) of image (a).

efficient filters [8]. A similar definition called black top-hat BTH(f) uses closing to enhance the black and narrow features:

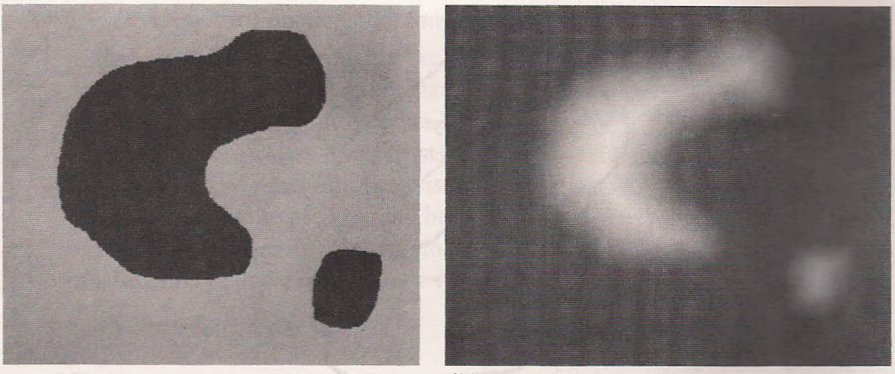
$$\mathrm{BTH}(f) = \varphi(f) - f$$

3. Distance Function

Let Y be a set of $\mathbf{Z}^2$. For every point y of Y, define the distance $d(y)$ of y to the complementary set Y^c (Figure 3):

$$\forall\, y \in Y, \qquad d(y) = \mathrm{dist}(y, Y^c)$$

where dist(y,Y^c) is the distance of y to the nearest point of Y^c.



(a) (b)

Figure 3. Distance function (b) of a set (a).

It can easily be shown that a section of d at level i is given by

$$X_i(d) = \{y : d(y) \geq i\} = Y \ominus B_i$$

where B_i is a disk of radius i.

This distance function is very helpful for segmenting binary objects, as shown later on.

4. Geodesy, Geodesic Distance

The geodesic transformations are very efficient in mathematical morphology. Starting from the notion of geodesic distance, one may define geodesic dilations and erosion and consequently, in geodesic spaces, the majority of the morphological transformations [9]. Here we introduce only the geodesic distance and two basic operators linked to this distance: the geodesic SKIZ (skeleton by zones of influence) and the reconstruction of a set from a marker.

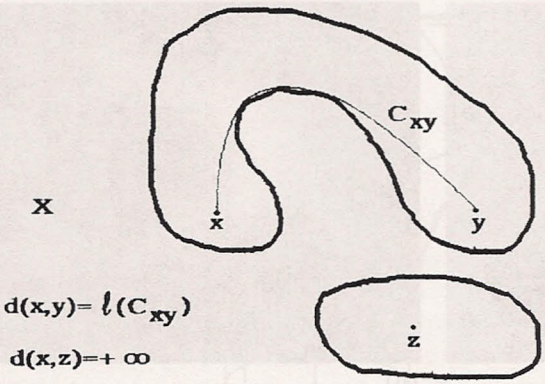
Let $X \subset \mathbf{Z}^2$ be a set, x and y two points of X. We define the geodesic distance $d_X(x,y)$ between x and y as the length of the shortest path (if any) included in X and linking x and y (Figure 4a).

Let Y be any set included in X. We can compute the set of all points of X that are at a finite geodesic distance from Y:

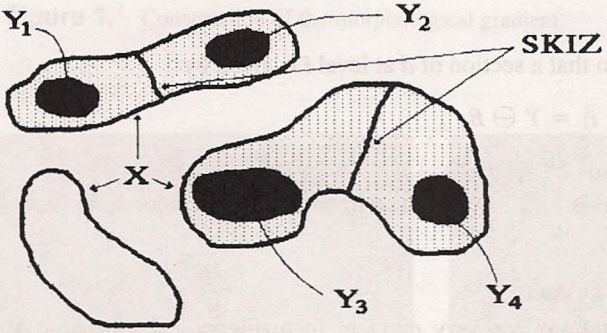
$$R_X(Y) = \{x \in X : \exists\, y \in Y,\ d_X(x,y) \text{ finite}\}$$

$R_X(Y)$ is called the X-reconstructed set by the market set Y. It is made of all the connected components of X that are marked by Y.

Suppose that Y is composed of n connected components Y_i. The geodesic zone of influence $z_X(Y_i)$ of Y_i is the set of points of X at a finite geodesic distance from Y_i and closer to Y_i than to any other Y_j (Figure 4b):



(a)



(b)

Figure 4. (a) Geodesic distance and shortest paths; (b) geodesic SKIZ of a set Y included in X.

$$z_X(Y_i) = \left\{ x \in X : \begin{array}{l} d_X(x,Y_i) \text{ finite} \\ \forall j \neq i,\ d_X(x,Y_i) < d_X(x,Y_j) \end{array} \right\}$$

The boundaries between the various zones of influence give the geodesic skeleton by zones of influence of Y in X, $\mathrm{SKIZ}_X(Y)$.

We shall write

$$\mathrm{IZ}_X(Y) = \bigcup_i z_X(Y_i)$$

and

$$\mathrm{SKIZ}_X(Y) = X \,/\, \mathrm{IZ}_X(Y)$$

where / stands for the set difference.

5. Geodesy for Functions: Reconstruction, Regional Extrema

Reconstruction. Introducing the geodesic transformations for the functions is not so easy because, on the one hand, there are many possible extensions of the binary operators and, on the other hand, the underlying geodesic distance is not obvious. Nevertheless, there exists a trick for extending the binary reconstruction to gray-tone images: it consists in using the sections of the functions. Indeed, any gray-tone picture may be considered either as a function f or as a pile of sections $X_i(f)$ (or $Z_i(f)$ as previously defined). Giving all the possible sections of a function f allows one to know for any point x the corresponding value $f(x)$:

$$f(x) = \max(i : x \in X_i(f))$$

or

$$f(x) = \min(i : x \in Z_i(f))$$

Consider two functions g and f and suppose that $f \leq g$. The corresponding sections of these two functions at level i are $X_i(g)$ and $X_i(f)$. This latter set is obviously included in the former one. For every level i, define a new set obtained by reconstructing $X_i(g)$ using $X_i(f)$ as a marker. It can be shown [5] that the new sets $R_{X_i(g)}(X_i(f))$ define a pile of embedded sections of a new function called the reconstruction of g by f (Figure 5) and denoted $R_g(f)$. In a similar way, the dual reconstruction of a function g by a function f (with $f \geq g$), denoted $R_g^*(f)$ is obtained by reconstructing the sections $Z_i(g)$ using $Z_i(f)$ as a marker (Figure 6).

As illustrated in Figure 6, the function f can be considered as a "wrap-up film" which packs the function g considered as a "parcel." The wrap-up film is of a type which contracts when heated. This contraction, however, occurs only in a horizontal direction, never in a vertical direction. The reconstruction and its dual transformation are clearly increasing. The reconstruction of g is always below the original function. Hence the transformation is antiextensive. Furthermore, the result remains unchanged if the reconstruction is repeated: the transformation is idempotent. It follows that the reconstruction is in fact a morphological opening [10]. The dual operation is a closing.

Minima, maxima of a function. Among the various features that can be extracted from an image, the minima and the maxima are of primary importance in the watershed transformation.

The set of all the points $\{x,f(x)\}$ belonging to $\mathbf{Z}^2 \times \mathbf{Z}$ can be seen as a topographic surface S. The lighter the gray value of f at point x, the higher the altitude of the corresponding point $\{x,f(x)\}$ on the surface.

The minima of f, also called regional minima, are defined as follows.

Consider two points s_1 and s_2 of this surface S. A path between $s_1(x_1,f(x_1))$ and $s_2(x_2,f(x_2))$ is any sequence $\{s_i\}$ of points of S, with s_i adjacent to s_{i+1}. A nonascending path is a path where

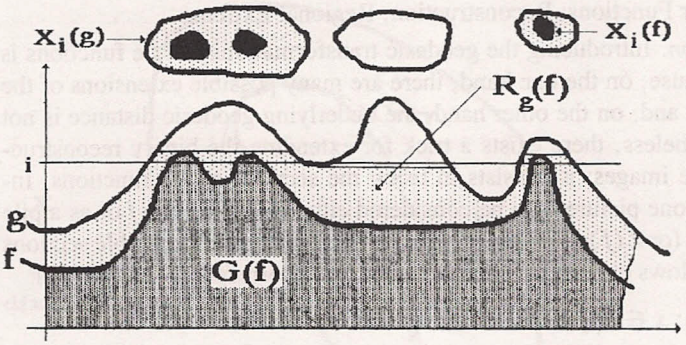


Figure 5. Reconstruction of a function g by a marker function f.

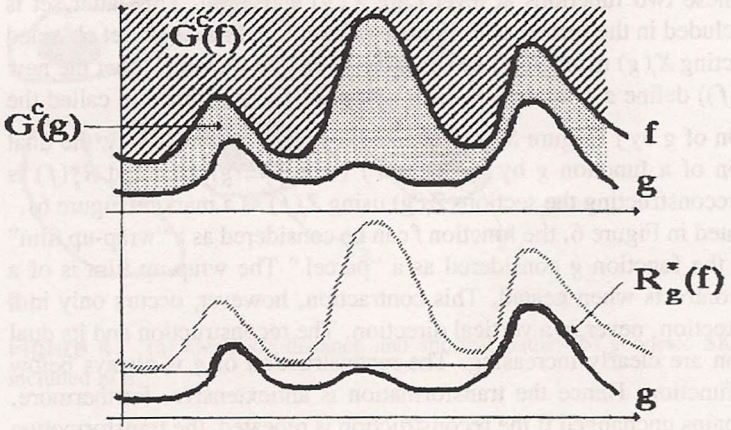


Figure 6. Dual reconstruction.

$$\forall\, s_i(x_i, f(x_i)),\ s_j(x_j, f(x_j)) \qquad i \geq j \Leftrightarrow \leq f(x_j)$$

This path is made of the concatenation of horizontal portions and of strictly descending ones.

A point $s \in S$ belongs to a minimum iff there exists no nonascending path stating from $s(x, f(x))$ and joining any point $s'(x', f(x'))$ of S such that $f(x') < f(x)$. A minimum can be considered as a sink of the topographic surface (Figure 7). The set $m(f)$ of all the minima of f is made of various connected components $m_i(f)$. A similar definition holds for the maxima $M(f)$.

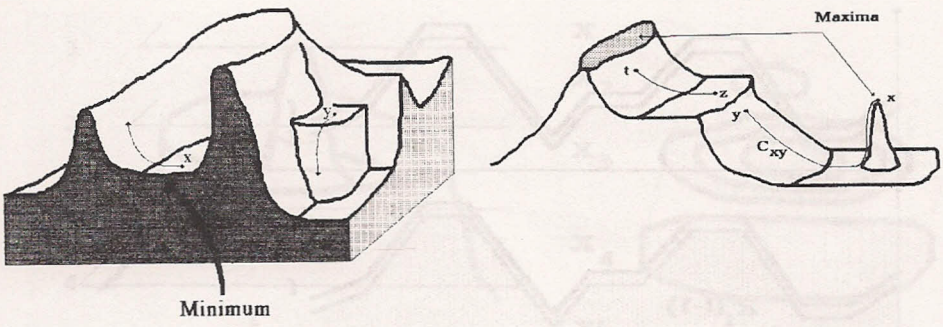


Figure 7. Minima and maxima of a function.

There exist various techniques for extracting the extrema of a function f. The most common one (but unfortunately one of the slowest ones) consists in using the reconstruction. It can be shown that [5]

$$k_{m(f)} = f - R_f(f - 1)$$

and

$$k_{M(f)} = R_f^*(f + 1) - f$$

where $k_{m(f)}$ and $k_{M(f)}$ are respectively the indicator functions of the minima and the maxima of f (Figure 8):

$$k_{m(f)}(x) = 1 \quad \text{iff } x \in m(f)$$
$$k_{m(f)}(x) = 0 \quad \text{if not}$$

6. Homotopy and Homotopic Transformations

Homotopy is a topological property of sets. Instead of defining homotopy in pure mathematical terms, let us simply give a practical definition: two sets X and Y are said to be homotopic if the first one can be superimposed onto the second one by means of continuous deformations. A transformation Φ is said to be homotopic if it transforms any set X into an homotopic set $\Phi(X)$ (Figure 9). A simply connected set will be transformed into a simply connected set, a set with one hole into a set with one hole, and so on. A typical example of homotopic transform is given by the skeleton of a set [11,12].

The extension of homotopy to functions is more difficult. In that case, it can be shown [11] that a homotopic transformation $\Phi(f)$ of a function f preserves the number and the relative positions of the extrema of f.

Another definition of homotopy for functions, more restrictive but easier to manipulate, can be used. Two functions f and g are said to be homotopic if, for any level i, the sections $X_i(f)$ and $X_i(g)$ are homotopic sets (Figure 10).

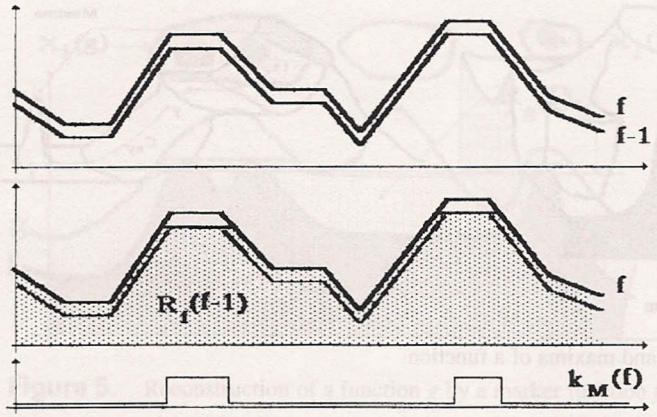


Figure 8. Maxima and reconstruction of a function f by $(f - 1)$.

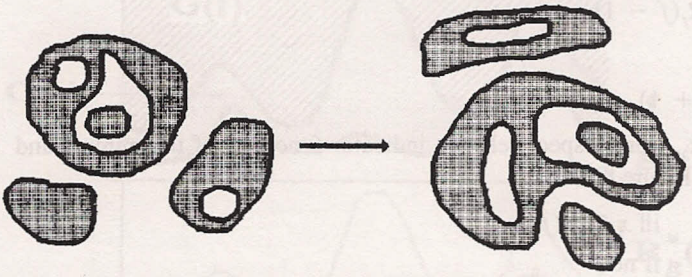


Figure 9. Example of homotopic transformation.

III. THE WATERSHED TRANSFORMATION

Let us introduce now one of the main tools used for segmentation in mathematical morphology: the watershed transformation [13]. After a didactic presentation as a flooding process, we shall explain its use for segmentation on a very simple example.

A. The Watershed Transformation

Consider again an image f as a topographic surface and define the catchment basins of f and the watershed lines by means of a flooding process. Imagine that we pierce each minimum $m_i(f)$ of the topographic surface S and that we plunge this surface into a lake with a constant vertical speed. The water entering through the holes floods the surface S. During the flooding, two or more floods coming

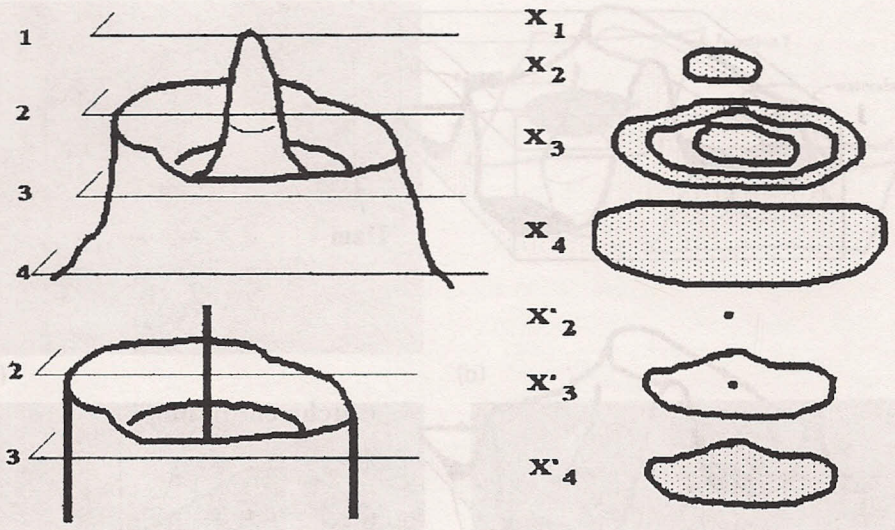


Figure 10. A restrictive definition of the homotopy for functions.

from different minima may merge. We want to avoid this event and we build a dam on the points of the surface S where the floods would merge. At the end of the process, only the dams emerge. These dams define the watershed of the function f. They separate the various catchment basins $CB_i(f)$, each one containing one and only one minimum $m_i(f)$ (Figure 11).

B. Use of the Watershed in Segmentation: A (Too) Simple Example

The application of the watershed to image segmentation will be shown through a very simple example: the segmentation of single dots in an image (radon gas bubbles in a radioactive material).

The dots in Figure 12a draw a topographic surface made of hollows with a roundish bottom. Each hollow has a unique bottom. The segmentation problem lies in finding the best contour of the bubbles.

A solution consisting of simply using a threshold is not sufficient because with a high threshold, the highest hollows are correctly detected, but the deepest ones are much too large. A lower threshold, while detecting correctly the deepest hollows, misses the higher.

Since absolute values cannot be used, we may try instead the variation of the gray-tone function, that is, its gradient (Figure 12c). The corresponding gradient image should present a volcano-type topography as depicted in Figure 12b. The contours of the radon bubbles therefore correspond to the watershed lines of the

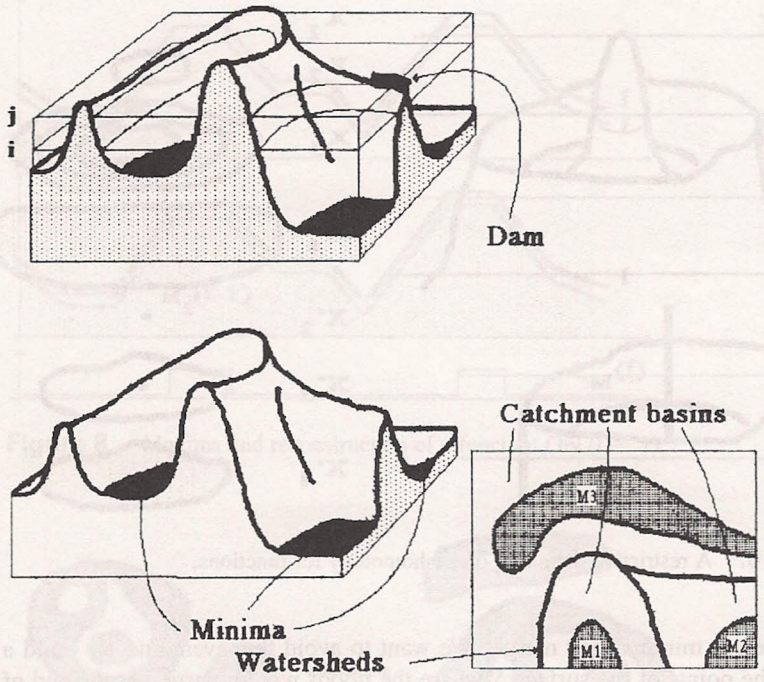


Figure 11. (a) Flooding of the relief and dam building; (b) catchment basins and watershed lines.

gradient image $g(f)$ (Figure 12d). In this gradient image, each dot of the original picture becomes a regional minimum surrounded by a closed chain of mountains. The bubble itself corresponds to a catchment basin of the gradient function, and the varying altitude of the chain of mountains expresses the contrast variation along the contour of the original dot.

C. Building the Watershed

Let us conclude this introductory example by a simple watershed algorithm which uses the basic morphological operators described in the first part.

The definition of the watershed transformation by flooding may be directly transposed by using the sections of the function f.

Consider (Figure 13) a section $Z_i(f)$ of f at level i, and suppose that the flood has reached this height. Consider now the section $Z_{i+1}(f)$. We see immediately that the flooding of $Z_{i+1}(f)$ is performed in the zones of influence of the connected components of $Z_i(f)$ in $Z_{i+1}(f)$. Some connected components of $Z_{i+1}(f)$ which are not reached by the flood are, by definition, minima at level $i + 1$.

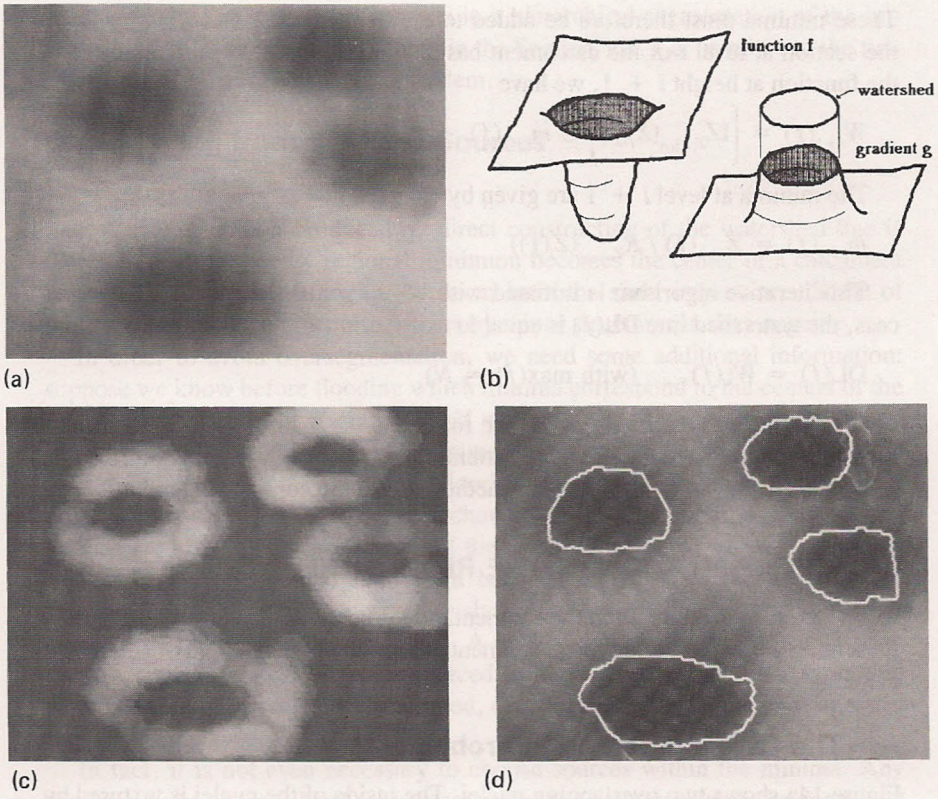


Figure 12. (a) Bubbles of gas in a radioactive material; (b) corresponding topographic surface of the initial function and of the gradient image; (c) morphological gradient; (d) watershed transform of the gradient image.

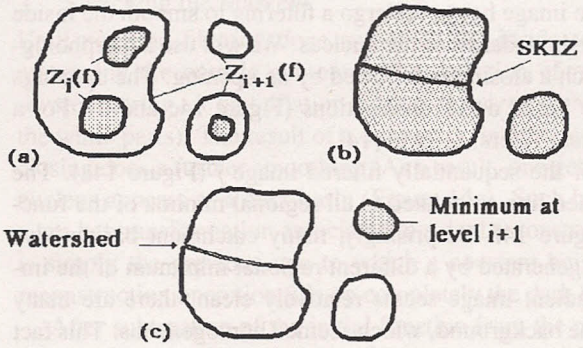


Figure 13. Watershed construction using a geodesic SKIZ.

These minima must therefore be added to the flooded area. Denoting by $W_i(f)$ the section at level i of the catchment basins of f and by $m_{i+1}(f)$ the minima of the function at height $i + 1$, we have

$$W_{i+1}(f) = \left[\mathrm{IZ}_{Z_{i+1}(f)}(X_i(f))\right] \cup m_{i+1}(f)$$

The minima at level $i + 1$ are given by

$$m_{i+1}(f) = Z_{i+1}(f) \,/\, R_{Z_{i+1}(f)}(Z_i(f))$$

This iterative algorithm is initiated with $W_{-1}(f) = \emptyset$. At the end of the process, the watershed line $\mathrm{DL}(f)$ is equal to

$$\mathrm{DL}(f) = W_N^c(f) \qquad (\text{with } \max(f) = N)$$

We shall discuss more deeply in the following the main groups of algorithms used for the watershed and focus our attention on some of them, but before doing so we must try to apply the previous method on a more complex example.

IV. SEGMENTATION IN THE REAL WORLD

The problems encountered in the segmentation process will be best illustrated by presenting a complete and typical segmentation problem in the field of automated cytology.

A. The Oversegmentation Problem

Figure 14a shows two overlapping nuclei. The inside of the nuclei is textured by the chromatin structure. Their outside is cytoplasm, which is textured itself. Any technique based on thresholding fails in this case. The importance of the nuclear texture obscures completely the gradient image (Figure 14b) and makes it difficult to discriminate between the contour lines of the nucleus and the chromatin patterns. For this reason the image has to undergo a filtering to smooth the inside texture while preserving the boundaries of the nucleus. We will use a morphological sequential filter in which a closing is followed by an opening. The openings and closings used here are based on reconstructions (Figure 14c and d). For a complete presentation of these filters, refer to [14].

Let g be the gradient of the sequentially filtered image f (Figure 14e). The construction of the watershed line associated to all regional minima of the function g is illustrated by Figure 14f. Surprisingly, many catchment basins have appeared. Each of them is generated by a different regional minimum of the image. And although the gradient image seems relatively clean, there are many regional minima, even in the background, which seemed homogeneous. This fact is general: the construction of the watershed line leads to severe oversegmentations. This may be amended by two types of methods. The first one [15] is pre-

sented below. The second [5] consists in a hierarchical segmentation of the image. This approach will be presented in the Section VII. Let us now see the first solution to the oversegmentation problem.

B. Flooding from Selected Sources

1. Description

The oversegmentation produced by direct construction of the watershed line is due to the fact that every regional minimum becomes the center of a catchment basin. Not all regional minima, however, have the same importance. Some of them are just produced by noise, others by minor structures in the image.

In order to avoid oversegmentation, we need some additional information: suppose we know before flooding which minima correspond to the centers of the nuclei and which to the background. If we come back to our flooding scheme, we will bore a hole only in these minima before immersing the relief. It is the only difference from the preceding algorithm; the flooding and the building of dams take place as previously. The catchment basins of the minima which are not pierced are filled up by overflowing of the neighboring catchment basin; as soon as the water reaches the saddle point between both basins, the water rushes through the pass and fills the previously empty basin (Figure 15). No dam is constructed between these two basins. A dam is constructed only for separating floods originating from different pierced minima. In the end, both spots and background will be covered by the flood, except for the divide line that separates them.

In fact, it is not even necessary to choose sources within the minima. Any region may be chosen. Nor is it necessary that the various markers be connected particles. It is sufficient that they share the same label. Two particles with the same label will be considered to belong to the same region and no dam will be erected between them, if their flooded areas happen to merge.

2. Searching the Markers

Until now the filtering done smoothed the inside texture of the nuclei, while preserving the outside contours. The detection of markers requires even more severe filtering. A first dilation reduces the sizes of the nuclei (a dilation enlarges the white parts). The result of a dilation is, in fact, an open set. A morphological closing does a further smoothing. As a result, one gets the function f, where each nucleus appears as a dark basin (Figure 14g). Such basins are easily detected by a top-hat transformation associated to a dual reconstruction. The marker function is simply the previous one to which a constant height h has been added. The reconstruction operation fills up completely the dark basins.

After subtraction of the initial function from the reconstructed one (in fact, it is a top-hat transformation, as the dual reconstruction is a closing), all nuclei appear as white domes. The final binary markers are obtained by thresholding

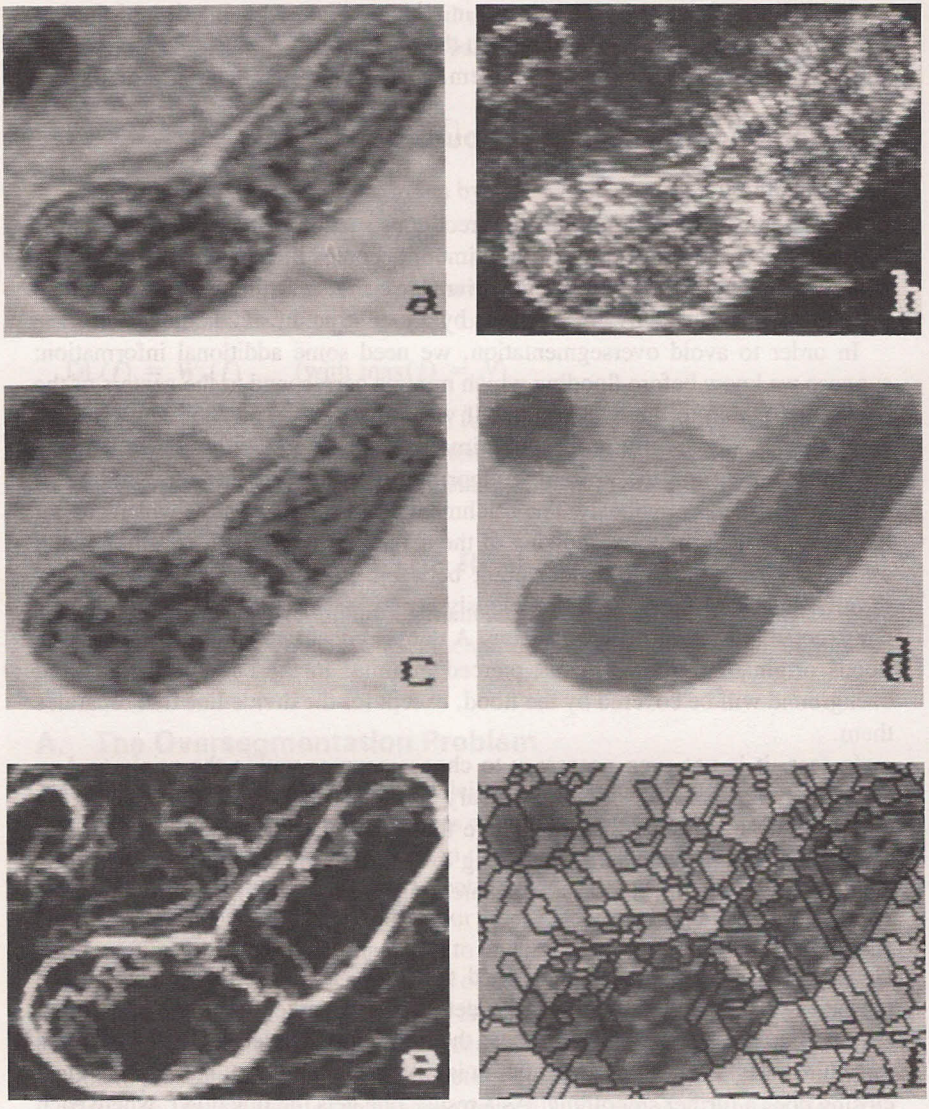
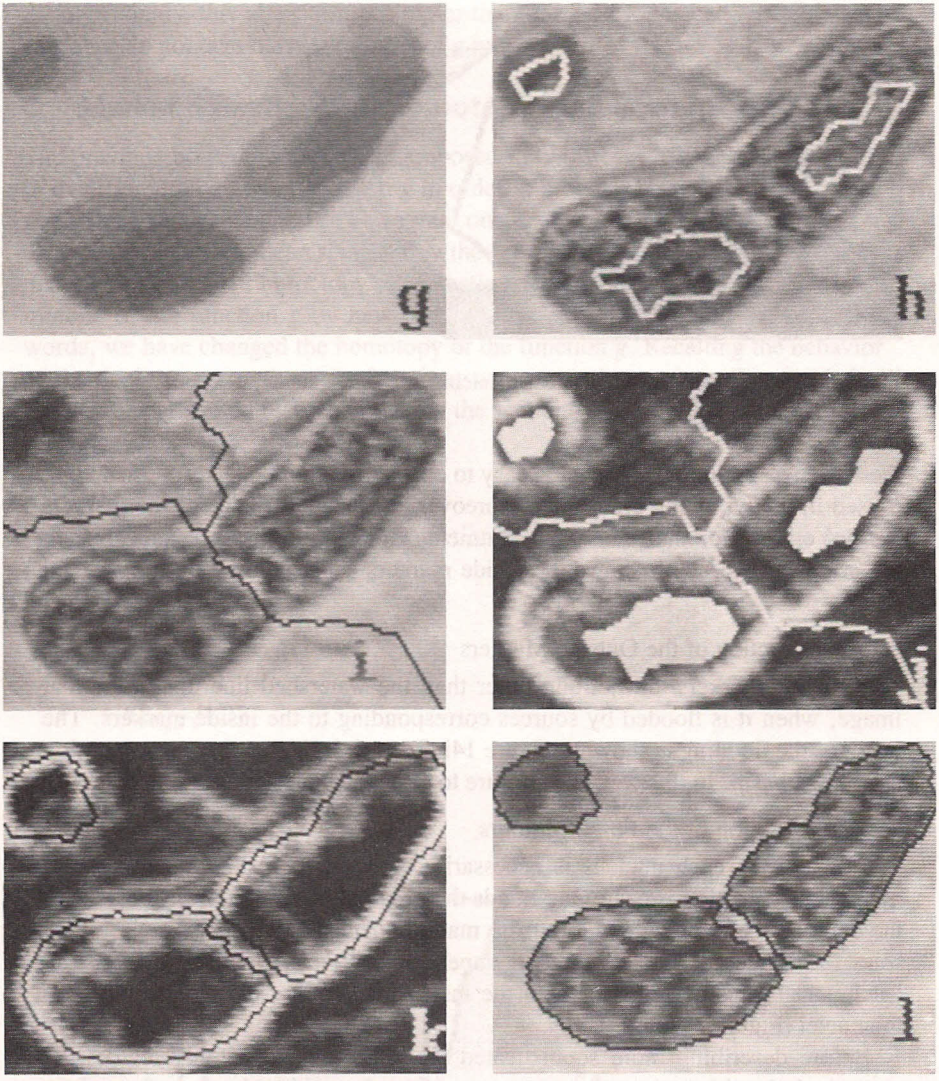


Figure 14. A typical sequence of segmentation. (a) Initial image of two overlapping nuclei. (b) Morphological gradient of the initial image. (c and d) Filtering of the original image. (e) Morphological gradient of the filtered image. (f) Watershed line of the gradient of the filtered image; the result is oversegmented. (g) After dilation and closing, each center of a nucleus appears as a dark basin. (h) Inside markers obtained by a top-hat transformation superimposed on the initial image. (i) Outside markers are the watershed lines of the initial image; the flooding sources are the inside markers. (j) Inside and outside markers superimposed on the gradient image. (k) Watershed of the gradient image with sources corresponding to the markers. (l) Resulting contour.


g
h
i
j
k
l

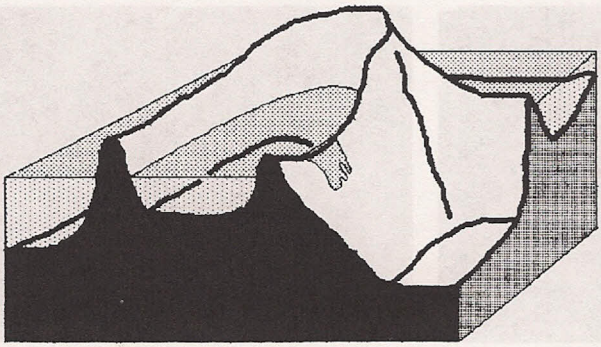


Figure 15. Overflow from a selected catchment basin to an adjacent one.

these domes; the threshold level is easy to choose, since it depends on the height h used in the wrapping algorithm. Moreover, the size and shape of the markers are not critical for the remaining treatment. Only their existence and location are critical. Figure 14h shows the inside markers superimposed on the original image.

3. Construction of the Outside Markers

The outside markers are nothing other than the watershed line of the original image, when it is flooded by sources corresponding to the inside markers. The result of the flooding is shown in figure 14i. To each inside marker corresponds a catchment basin. In this way, we are sure to select a connected outer marker.

4. Construction of the Final Contours

The contour of each nucleus is necessarily between its inside and its outside marker. Its detection is easy. One floods the gradient image obtained previously; the sources are the inside and outside markers detected above. Figure 14j presents the inside and outside markers superimposed on the gradient image. The catchment basins corresponding to the inside markers are the binary masks of the nuclei (Figure 14k and l).

Before describing more sophisticated algorithms, let us simply rewrite the watershed algorithm given above when we introduce this selection of markers. This algorithm can be written as follows.

If $W_i(g)$ is the section at level i of the new catchment basins of g, we have

$$W_{i+1}(g) = \mathrm{IZ}_{(Z_{i+1} \cup M)}(W_i(g))$$

with

$$W_{-1}(g) = M, \qquad \text{marker set}$$

Surprisingly, this algorithm is simpler than the pure watershed algorithm because we do not take the real minima of g into account.

C. Marker Selection and Homotopy Modification

The previous procedure can be decomposed in two steps. The first one consists in modifying the gradient function g in order to produce a new gradient g'. This new image is very similar to the original one, except that its initial minima have disappeared and have been replaced by the set M (Figure 16). This image modification is also called homotopy modification. In fact, we have replaced the old minima of the function g by new ones corresponding to the markers; in other words, we have changed the homotopy of the function g. Recalling the behavior of the dual reconstruction of a function, we can easily see from Figure 16 that this can be performed by reconstructing the sections of g with the markers M.

We have

$$\forall i, \quad Z_i(g') = R_{Z_i(g) \cup M}(M)$$

If we denote by k_M the indicator function of the markers and by $i_{\max}$ the maximum value of g, we can write

$$k' = i_{\max}(1 - k_M)$$

and then

$$g' = R^*_{\mathrm{Inf}(g,k')}(k')$$

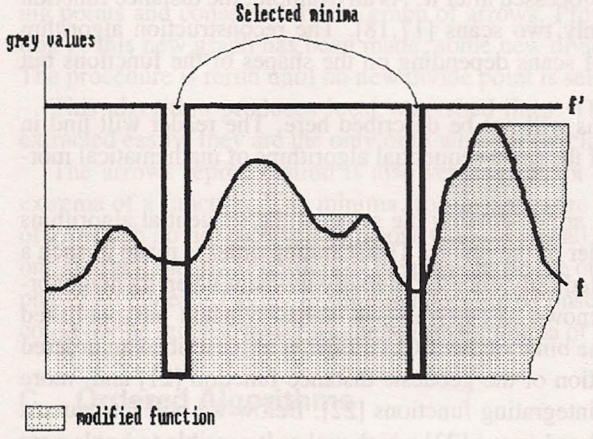


Figure 16. Principle of the homotopy modification of a function by a set of selected markers.

The second step simply consists in performing the watershed of the modified gradient g'.

V. ALGORITHMS OF WATERSHED

A. Review of the Different Classes of Algorithms

The watershed algorithms can be divided in two groups. The first group contains algorithms which simulate the flooding process. The second group is made of procedures aiming at direct detection of the watershed points. Each group of algorithms can subsequently be divided into three classes: parallel algorithms, sequential ones, or ordered algorithms.

An algorithm is parallel if the neighboring points of the point to be transformed take all their values in the original image. The algorithm is said to be parallel, because the result is independent of the order in which the points are transformed. As a matter of fact, all points could be transformed in parallel.

In a sequential (also called recursive) algorithm, the newly computed value of a point will serve as argument for the transformation of its not yet transformed neighbors. The result of the transformation depends completely on the scanning order. Generally, for simplifying the access to the image memories, one adapts forward and inverse raster scanning. Rosenfeld and Pfaltz showed the equivalence between parallel and sequential algorithms [16].

The sequential algorithms are generally much faster than the parallel algorithms. In a parallel algorithm, the value of a point has an influence only on its neighbors. In a sequential algorithm the value of a point may have an influence on the values of all points processed after it. As an example, the distance function of a binary set requires only two scans [17,18]. The reconstruction algorithm needs a variable number of scans depending on the shapes of the functions but not on their size.

The recursive algorithms will not be described here. The reader will find in [19] an extensive review of the main sequential algorithms of mathematical morphology.

The ordered algorithms are essentially the same as the sequential algorithms except for the scanning order of the points. The scanning order is made in such a way that each point is visited only once, at the very moment when its neighborhood is sufficiently well known to determine its value. Vincent has published many such algorithms in the binary case [20]. Verwer et al. described an ordered algorithm for the construction of the geodesic distance function [21] and, more recently, an algorithm for integrating functions [22]. Below we will introduce a data structure called an ordered queue [23] which makes it possible to implement in a quite natural way a series of ordered algorithms. We will describe the implementation for the reconstruction transformation and for the construction of the watershed line. The algorithms described in the previous section belong to the

first group and are parallel; they simulate the flooding of the topographic surface drawn by f.

Before presenting the ordered algorithms which also belong to the first group, let us briefly describe another algorithm belonging to the second group and based on the arrows representation of a function f [5].

B. A Brief Introduction to a Second Group Algorithm

From $f : \mathbf{Z}^2 \rightarrow \mathbf{Z}$, we may define an oriented graph whose vertices are the points of $\mathbf{Z}^2$ and with edges or arrows from x to any adjacent point y iff $f(x) < f(y)$ (Figure 17).

The definition does not allow arrowing of the plateaus of the topographic surface. This arrowing can be performed by means of geodesic dilations. The operation is called the completion of the arrows graph. Moreover, in order to suppress problems due to the fact that a watershed line is not always of zero thickness, a more complicated procedure called overcompletion is used, which leads to double arrowing for some points. Then, starting from this complete graph (overcompleted), we may select some configurations which, locally, correspond to divide lines. These configurations are represented in Figure 18 for the 6-connectivity neighborhood of a point on a hexagonal grid (up to a rotation).

Any point receiving arrows from more than one connected component of its neighborhood may be flooded by different lakes. Consequently, this point may belong to a divide line. In a second step, the arrows starting from the selected points must be suppressed. These points, in fact, cannot be flooded, so they cannot propagate the flood. In doing so, we change the arrowing of the neighboring points and consequently the graph of arrows. Provided that the overcompletion of this new graph has been made, some new divide points may then appear. The procedure is rerun until no new divide point is selected (Figure 19).

This algorithm produces local watershed lines. The true divide lines can be extracted easily; they are the only ones which form closed curves.

The arrows representation is also very useful for detecting very quickly the extrema of a function. The minima, for instance, are the connected components of $\mathbf{Z}^2$ which do not receive any arrow. One detects all plateaus of the function on one side and the lower borders of the plateaus on the other side. Then, in a second phase, all plateaus having lower neighbors are reconstructed. The plateaus which could not be reconstructed are the regional minima of the image.

C. Ordered Algorithms

Leaving the algorithms of the second group, let us come back to those of the first group using an ordered queue. We shall describe mainly the implementation of the dual reconstruction and of the construction of the catchment basins. We shall first introduce a data structure called ordered queue (OQ). Its principal merit is

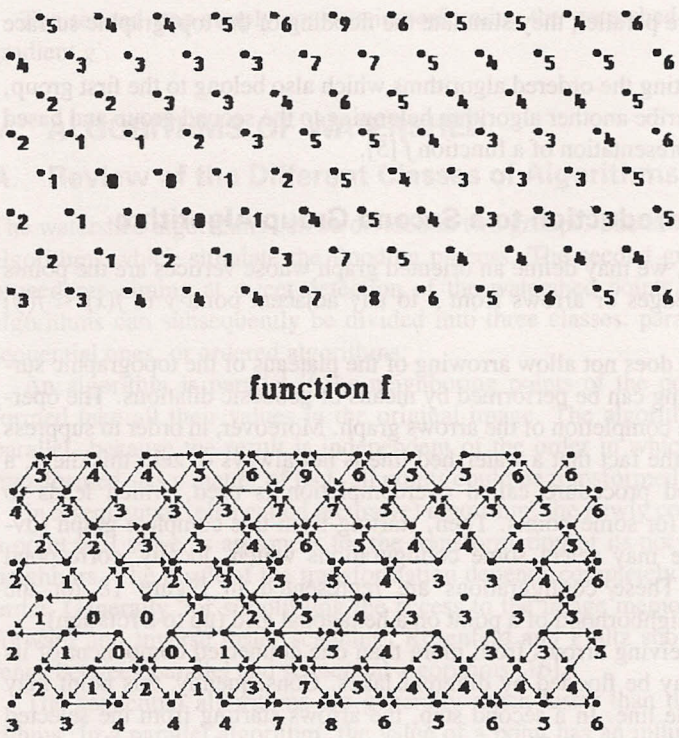


Figure 17. (a) Function f and (b) its corresponding graph of arrows.

to facilitate the storage of points in any order and their retrieval in the order of flooding. For this reason, this structure is at the base of an elegant optimal implementation of the reconstruction operation and watershed line.

1. The Ordered Queue

A hierarchical ordering relation in flooding. During the flooding of a topographic surface, there appears a dual order relation between the pixels (we consider here the flooding with sources placed at the regional minima of the function). It is clear that a point x is flooded before a point y if y is higher than x on the relief. This constitutes the first level of the hierarchy. It is simply the order relation between the gray values. A second order relation occurs on the plateaus. Let X be a plateau at an altitude h. Before X begins to be flooded, all neighboring

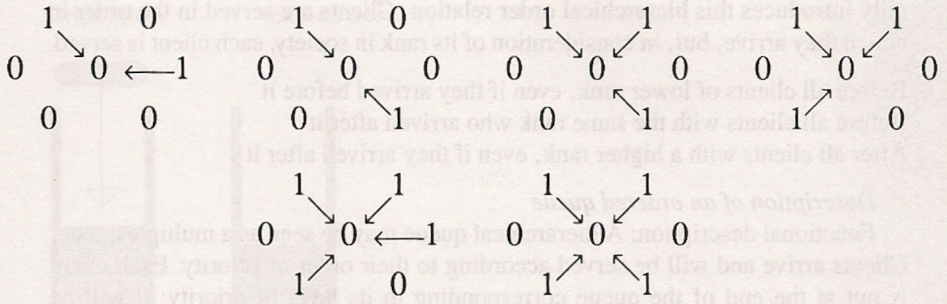
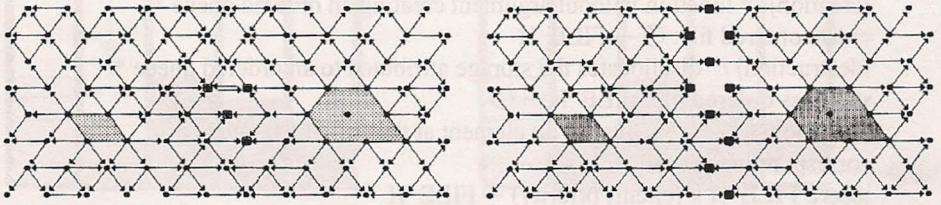


Figure 18. Configurations of arrows corresponding to possible divide points (hexagonal grid).



Selection of primary points **Final result**

Figure 19. Watershed by arrowing: (a) primary divide points (saddle points); (b) final result.

points of X with a lower altitude than h have been flooded. One supposes that the flooding of the plateau is not instantaneous but progressive. The flood progresses inward into the plateau with uniform speed. The first neighbors of already flooded points are flooded first. Second neighbors are flooded next, etc. This introduces a second order relation among points with the same altitude, corresponding to the time when they are reached by the flood. If two points x and y belong to the same plateau X, of height h, x will be reached by the flow before y if the geodesic distance within the plateau X to the points of lower altitude is smaller for x than for y. In the next section we show that an ordered queue natu-

rally introduces this hierarchical order relation. Clients are served in the order in which they arrive, but, in consideration of its rank in society, each client is served

Before all clients of lower rank, even if they arrived before it
Before all clients with the same rank who arrived after it
After all clients with a higher rank, even if they arrived after it

Description of an ordered queue

Functional description: A hierarchical queue may be seen as a multiple queue. Clients arrive and will be served according to their order of priority. Each client is put at the end of the queue corresponding to its level of priority: it will be served after all clients with the same priority who arrived before it. Only one client may be served at a time. Once the queue of a given priority is empty, it is suppressed. If a client with high priority arrives after the suppression of the queue to which it belongs, it will be put in the queue of highest priority still existing.

The ordered queue is organized in such a way that it is possible to know whenever a client extracted from the queue has a lower priority than the previous client. The functional specification of an ordered queue is the following:

{creation} /* function without argument creating an ordered queue */
create ordered file: ⓘ → FILE_H
{destruction} /* disallocates the storage attributed to an ordered queue */
suppress ordered file: FILE_H → ⓘ
{insertion} /* inserts an element at the end of the queue of corresponding priority */
insert: FILE_H x (client, priority) → FILE_H
{serve} /* gives the address and priority of the client with the highest priority, who arrived first */
serve: FILE_H → element

Illustration of the possible actions: Figure 20 shows how a simple queue works. We have represented the queue as a cylinder and the clients as coins in the cylinder. Each arriving client is put on the top of the cylinder. An opening at the base of the cylinder permits the removal of the client who arrived first.

Figure 21a–d show how an ordered queue works. It can be seen in Figure 21a that an ordered queue is in fact a series of simple queues. Each simple queue is assigned a level of priority. In the drawings the priority is represented as a gray value; the darkest gray values correspond to the highest priorities. In our example, we have five levels of priority. All cylinders are open at the top, which means that at any moment it is possible to introduce a client of any priority in the queue. On the contrary, only the queue with the highest priority has an opening at its basement. Figure 21b shows the extraction of an element of the structure: it is the client who arrived first in the queue of highest priority still existing in the

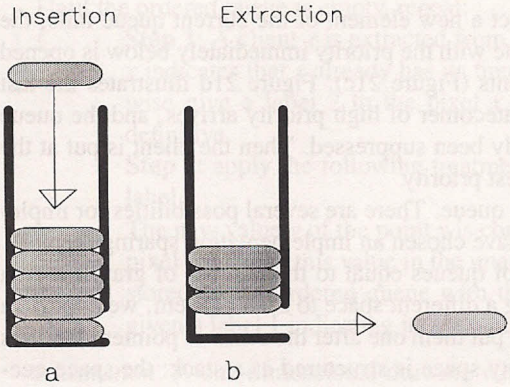


Figure 20. Mechanism of a simple queue.

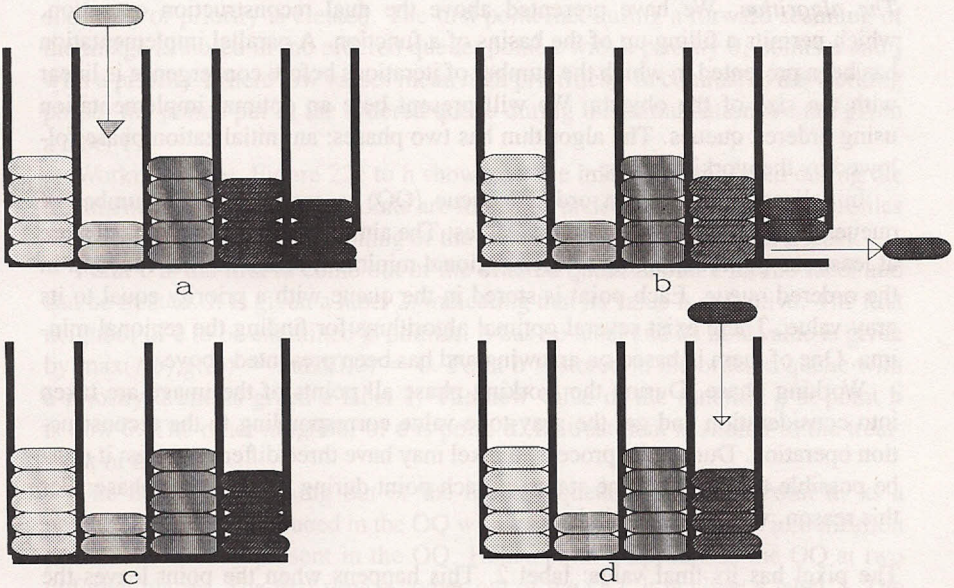


Figure 21. Principle of an ordered queue: (a) the ordered queue; (b) extraction of an element of highest priority; (c) the number of simple queues is reduced; (d) treatment of a highest-priority element when the corresponding queue has been suppressed.

structure. If the attempt to extract a new element of the current queue fails, the queue is suppressed and the queue with the priority immediately below is opened for extraction of the next elements (Figure 21c). Figure 21d illustrates the last feature of an ordered queue: a latecomer of high priority arrives, and the queue with the same priority has already been suppressed. Then the client is put at the end of the current queue of highest priority.

Implementation of an ordered queue. There are several possibilities for implementing an ordered queue. We have chosen an implementation sparing memory. One has to represent a number of queues equal to the number of gray levels in the image. Rather than allocating a different space to each of them, we allocate a common space to all of them and put them one after the other. A pointer identifies the end of each queue. The empty space is structured as a stack: the space necessary to store a newcomer is taken at the top of the stack; conversely, the space liberated by a client leaving the ordered queue is returned to the stack.

2. The Reconstruction Algorithm

The algorithm. We have presented above the dual reconstruction operation, which permits a filling up of the basins of a function. A parallel implementation has been presented in which the number of iterations before convergence is linear with the size of the objects. We will present here an optimal implementation using ordered queues. The algorithm has two phases: an initialization phase followed by the working phase.

Initialization phase. An ordered queue (OQ) is created with a number of queues equal to the number of gray values. The aim of the initialization is to store at least one point belonging to each regional minimum of the marker function in the ordered queue. Each point is stored in the queue with a priority equal to its gray value. There exist several optimal algorithms for finding the regional minima. One of them is based on arrowing and has been presented above.

Working phase. During the working phase all points of the image are taken into consideration and get the gray-tone value corresponding to the reconstruction operation. During the process, a pixel may have three different states; it must be possible to recognize the status of each point during the working phase. For this reason, we use three labels:

The pixel has its final value: label 2. This happens when the point leaves the ordered queue.
The pixel has been stored in the ordered queue but has not been assigned its final value yet: label 1.
The pixel has never been taken into consideration: no label.

f is the initial function (also called "parcel" in the illustration), g the function (above f) used as a marker for the dual reconstruction (named "film"). The treatment goes as follows:

Until the ordered queue is empty, repeat:

{ **Step 1**: A client x is extracted from the ordered queue. If the label of x indicates that x already has its final value, start again step 1. Otherwise give a label 2 to the pixel x, indicating that its value is now definitive.

Step 2: apply the following treatment to any neighbor y of x without label:

The new value v of the point y is computed: $v = \max(g(x),f(y))$. The pixel y is given this value in the image g. The address of the pixel y is stored in the ordered queue with the priority v. The pixel y is also given a label 1 indicating its presence in the queue. }

Illustration. A one-dimensional drawing will illustrate the way the algorithm works. Figure 22a represents a profile of the functions f (parcel) and g (film). The function f is hatched; the function g is above f and is indicated with bold lines. The regional minima of the function g are indicated below the function.

Initialization. The function f has six gray levels. Hence an ordered queue with six levels of priority is created. The first point met during a forward scanning of the image is stored in the ordered queue: point **c** with a priority 0, points **a** and **j** with a priority 1 (here low values mean high priorities). In contrast to the working phase, the points put in the ordered queue during the initialization are not given a label 1.

Working phase. Figure 22c to h show how the image g is modified during the treatment. The labels of the points are indicated under the corresponding profiles of the functions. At the beginning of the working phase, no point has a label.

Point **c** is the first to come out of the ordered queue. Point **c** has no label and can be treated. It is given a label 2, indicating that its value is definitive. The first neighbor of **c** to be examined is point **b**. **b** has no label and its new value is given by $\max(f(b),g(c)) = \max(0,0) = 0$. Point **b** is stored in the ordered queue with a priority 0 and is given a label 1. The new value of the function g at point **b** is now 0. The other neighbor of **c** is point **d**. Its treatment is similar to the treatment of **b**.

The next point coming out of the ordered queue is point **b**. Point **a**, as a neighbor of **b**, is introduced in the OQ with a priority 0. But **a**, as an initialization point, was already present in the OQ. Hence point **a** appears in the OQ at two different places. The first time **a** comes out of the OQ, it is given a label 2. The second time it will not be further processed.

The processing of point **d** introduces point **e** in the OQ with a priority 4. This is represented in Figure 22d.

The treatment of **a** consists in giving it the label 2. As a boundary point, point **a** introduces no other point in the OQ. The queue of priority 0 is now empty and is removed from the structure. The next point comes out of the queue of priority 1; it is point **a**, which has already a label 2. Point **a** is skipped and one proceeds

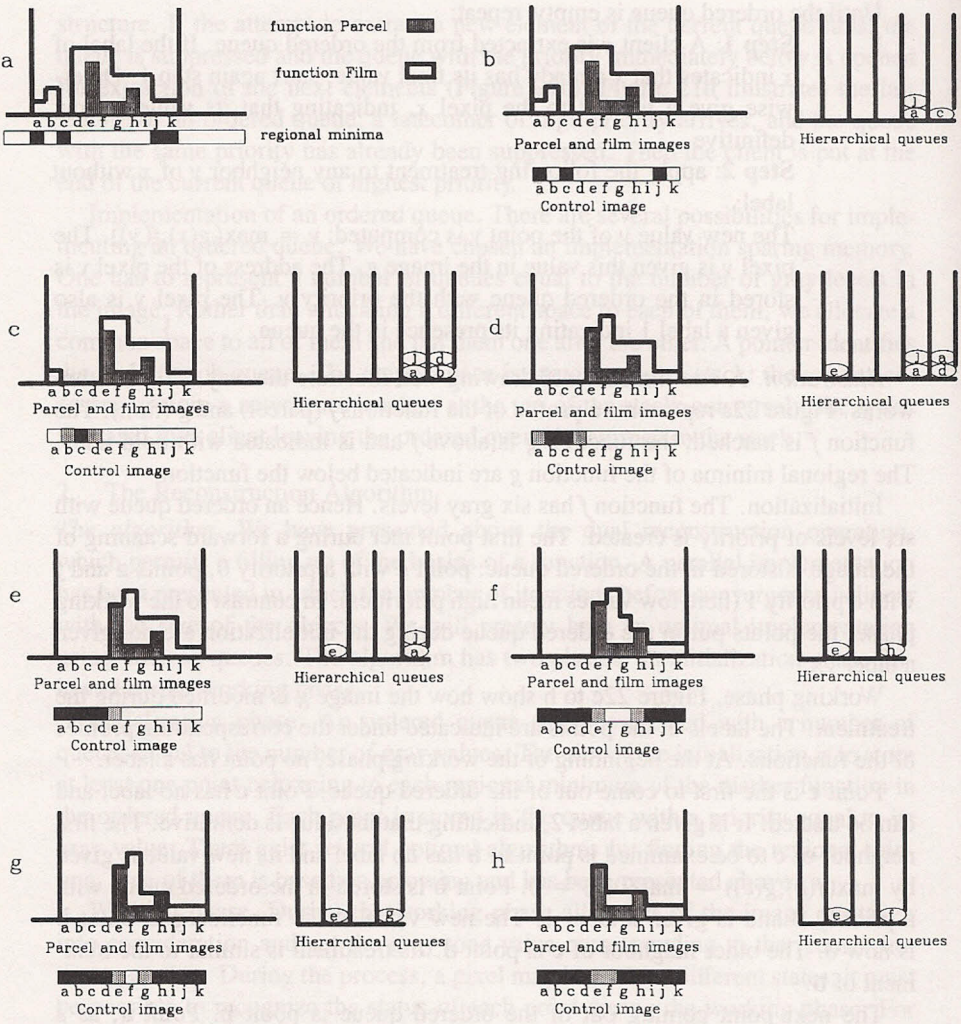


Figure 22. Illustration of the dual reconstruction by an OQ algorithm (see text).

by treating point **j**. The treatment of **j** introduces points **i** and **k** in the OQ with a priority 1. After the treatment of points **i** and **k**, point **h** is introduced in the OQ with a priority 2. The queue of priority 1, being empty, is suppressed. The function g has now the shape indicated in Figure 22f.

The treatment proceeds in this way (Figure 22g) until the result indicated in Figure 22h is obtained. The last two points **c** and **f** present in the OQ have all

their neighbors labeled; they are not replaced when they leave the OQ. At this moment, the OQ is empty and the treatment is finished.

The points are treated in an order proportional to their gray values. Each point is processed only once. In this sense, the algorithm is optimal.

3. The Watershed Algorithm

General presentation. The input is now a gray-tone function f to be flooded and a set of markers M, which serve as sources for the flooding. If the markers are the regional minima of the image f, then the result is the plain watershed line associated with the relief f. If it is not the case the result is the watershed line of a function $f' = R^*_{\mathrm{Inf}(f,k')}, k'$ being the function defined in Section IV.C.

The use of an ordered queue makes it possible to flood directly from a set of markers without doing the reconstruction operation. Indeed, the picturesque presentation of the flooding in Figure 15 will be faithfully simulated.

The markers are identified by labels. Each region will keep the label of the marker which has been the source of the flood. A marker may have several distinct connected components as long they share the same label.

There exist two versions of the algorithm. In the first version, the catchment basins touch each other, without any frontier between them. In the second version such frontiers are generated. Only the first version will be presented in detail.

The algorithm. An initialization phase is followed by a working phase.

Initialization. An ordered queue is created with as many priority levels as there are gray tones in the image f.

A boundary point of a marker belongs to a marker and has in its neighborhood a point outside a marker. All boundary points of the markers are entered in the ordered queue; the value of each point in the image f determines the priority level in the ordered queue.

Working phase. An image g is created by labeling the markers M. The treatment follows:

Until the ordered queue is empty, repeat:

{ A client x is extracted from the ordered queue. To each neighbor y of x having no label in the image g the same treatment is applied:
- the point y is given the same label as x in the image g.
- the point y is stored in the ordered queue; its value in the image f determines its priority level in the ordered queue. }

Illustration. The series of Figure 23a–h illustrate how the algorithm works. The left part of the figure presents on the top the topographic surface and below the zone which has been flooded. Each flooded zone bears the label of the source from which it has been flooded. The content of the ordered queue is represented in the right part of each figure.

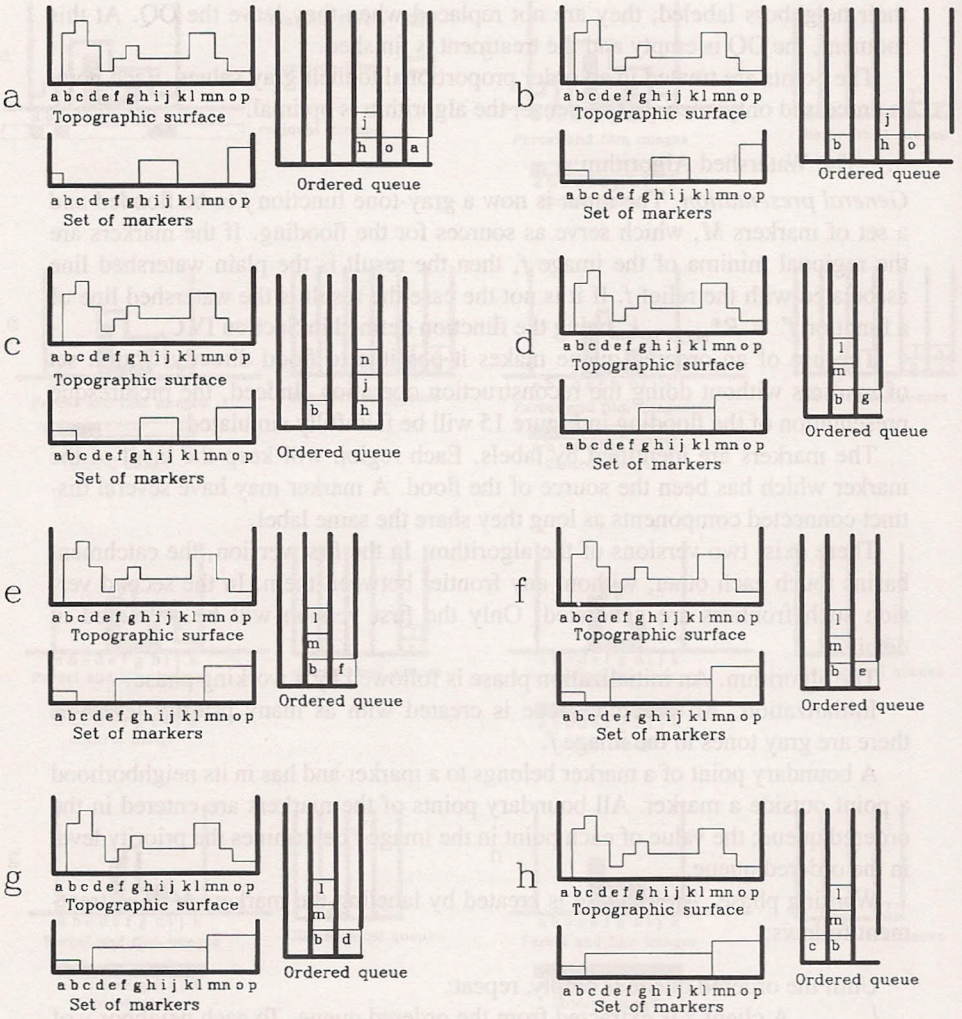


Figure 23. Watershed line construction using an OQ algorithm for simulating the flooding (see text).

Initialization. An ordered queue is created with six levels of priority corresponding to the five gray tones of the topographic surface.

The inside boundary points are stored in the ordered queue with a priority corresponding to their altitude on the topographic surface: point **a** with priority 0, point **o** with priority 1, and points **h** and **j** with priority 2. The resulting ordered queue is shown in Figure 23a.

Working phase. Point **a** is the first point to leave the ordered queue. Its only neighbor is **b**. Point **b**, having no label, takes the label 1 from **a** and is put into the ordered queue with a priority equal to its altitude, i.e., 4. The queue of level 0 is now empty and is suppressed. The state of images and queues is illustrated by Figure 23b.

The next point leaving the ordered queue is **o**. Its left neighbor having no label gets the label 3 from **o** and is stored in the queue of priority 2. The right neighbor of **o** is **p**; **p** already has a label and is not further processed (Figure 23c). The queue of priority 1, being empty, is suppressed.

The treatment of the points belonging to the queue of priority 2 proceeds as follows:

Point **h** gives its label 2 to its neighbor **g**: **g** enters the ordered queue with priority 3.

Point **j** gives its label 2 to its neighbor **k**: **k** enters ordered queue with priority 2.

Point **n** gives its label 3 to its neighbor **m**: **m** enters the ordered queue with priority 4.

Point **k** gives its label 2 to its neighbor **l**: **l** enters the ordered queue with priority 4.

The queue of priority 2 is now empty and is suppressed. The state of images and queues is illustrated in Figure 23d.

The next point to be treated is **g**. Its only neighbor without a label is **f**. But **f** has an altitude equal to 3 and should be put in the queue of priority 2. Yet, this queue has just been suppressed. Point **f** will then be put in the queue with the highest priority still existing, in our case the queue of priority 3. Simultaneously, point **f** is given the label 2. The flooding coming from the source (**hij**) will fill up the catchment basin associated with the minimum **e**, where no marker was placed.

The flooding of this neighboring catchment basin continues with the treatment of point **f**. Point **e** is introduced in queue 3, despite the fact that its altitude is 1 (Figure 23f). After the treatment of the points **e** and **d**, queue 3 is now empty and is suppressed (Figure 23g and h). The next points to leave the OQ are successively points **b**, **m**, **l**, and **c**. All their neighbors already having labels, their treatment introduces no new points in the OQ. Thus the queue is completely emptied when the last point, **c**, leaves it. This achieves the flooding. Each point of the image has been assigned to the region from which the flood came first.

Discussion. The implementation of the watershed line we have described is the simplest and fastest using an ordered queue. A slight modification of the rules which affect each point to the catchment basins leads to several variants. As may be seen in Figure 23, the algorithm does not produce frontiers between catchment basins: each point in the field belongs to a catchment basin.

Another version of the watershed ordered algorithm produces a frontier with a thickness of one pixel point. Both algorithms share the following features:

Each point being considered only once during the treatment phase, the algorithms are indeed optimal.

The flooding is done according the order relation induced by the ordered queue and analyzed above under "The Watershed Algorithms."

VI. EXAMPLES OF SEGMENTATIONS

Three examples of segmentation are described in this section. Each one has been chosen to illustrate a particular topic of this methodology. The first example, the electrophoresis gel segmentation, although not very complex shows that different choices of markers may lead to different results. The second example, the overlapping grains separation, is a binary application of the watershed segmentation. In this case, the criterion used for segmenting objects is based not on their gray values but on their relatively convex shapes. The third example, finally, is more complex. The objects to be segmented are facets in a cleavage fracture in steel. It shows that, despite the fact that the markers are difficult to obtain, once they are defined, the tasks consisting in comparing the facets or defining their neighborhood relationships become easier.

A. Segmentation of Electrophoresis Gels

This first example consists of contouring blobs of proteins in an electrophoresis gel (Figure 24a). This problem seems to be easier than the nuclei segmentation presented above and, in fact, a similar approach is used.

The initial image is filtered. An alternate sequential filter is applied. The minima of the filtered image are the markers of the blobs (Figure 24b). We must also define a marker for the background. In order to get a connected marker surrounding the blobs, we use, as we did for the cells, the watershed of the initial filtered image (Figure 24c). From this, we obtain our set of markers M (Figure 24d). Finally, the watershed of the modified gradient image is performed. The result is given in Figure 24e.

It is clear in this example that the final segmentation depends on the selection of the minima of the initial function as blob markers. If some blobs do not correspond to minima (as is sometimes the case), they will not be contoured correctly. Moreover, using a connected marker for the background induces, by construction, each detected blob to be surrounded by a simple closed arc and that there are no touching blobs.

But, if we use another marker for the background, the result will be different. To demonstrate this, let us choose as background marker the maxima of the initial filtered image (Figure 25a). This marker is not connected, and the watershed

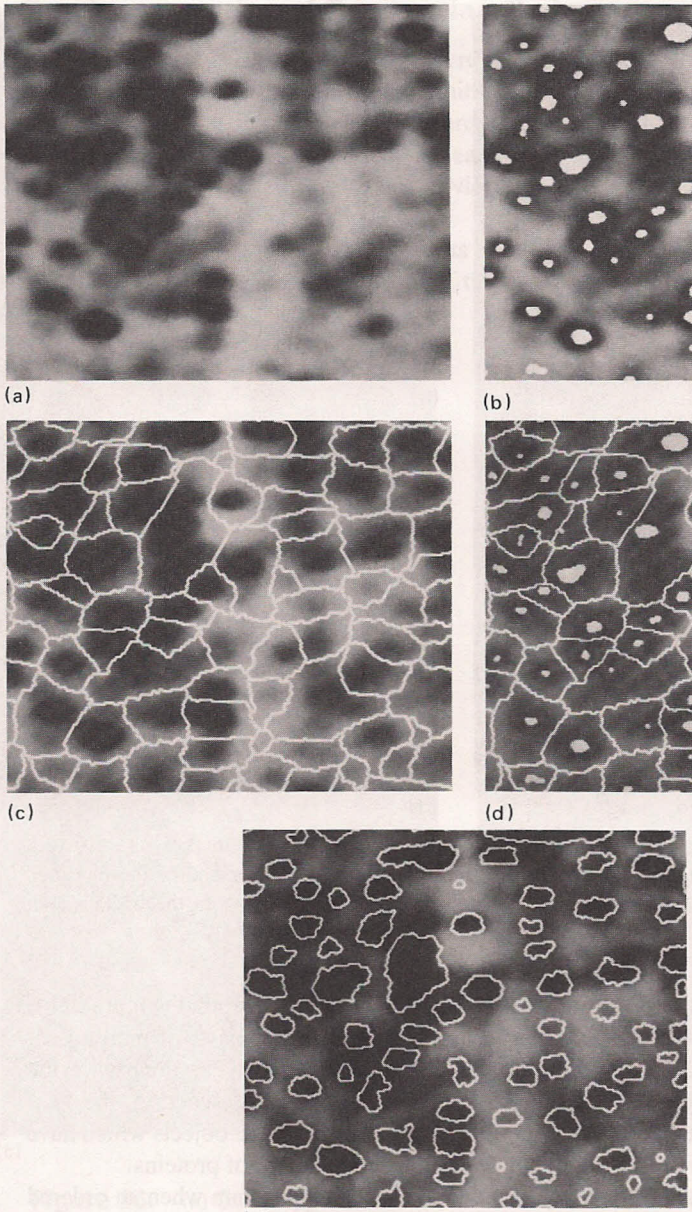


Figure 24. (a) Electrophoresis gel; (b) minima of the filtered image marking the blobs; (c) watershed of the filtered image used as background marker; (d) set of selected markers; (e) final segmentation.

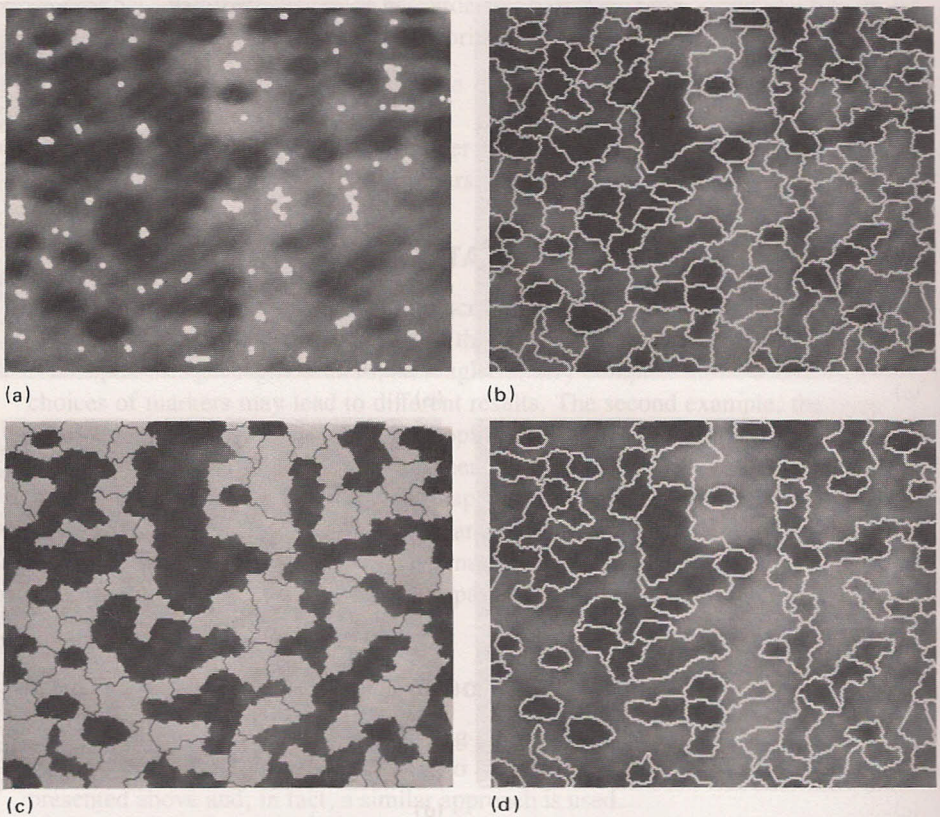


Figure 25. Segmentation obtained when the background marker is changed. (a) Background markers (minima of the filtered image); (b) watershed of the modified gradient; (c) reconstruction of the catchment basins corresponding to the background; (d) final result.

segmentation produces many catchment basins in the background (Figure 25b). Suppressing this oversegmentation is straightforward; it consists of merging all the basins marked by the maxima (Figure 25c). If the same is done with the catchment basins corresponding to the minima of the initial function, the final segmentation (Figure 25d) is rather different. In that case, the objects which have been detected are not the individualized blobs but the heaps of proteins.

Note that the use of a nonconnected marker is not a problem when an ordered algorithm is used. In such a case, all the connected components of the marker have the same label.

B. Segmentation of Overlapping Grains

This example presents another case where the watershed line is most useful: the separation of overlapping grains. The initial picture (Figure 26a) represents coffee grains. This picture can easily be thresholded and it may be seen from their shape that many grains overlap or touch each other. To segment them, no contrast criterion can be used because there is obviously no visible boundary between two overlapping grains.

The solution of the problem consists in using the distance function of the binary set (Figure 26b). The maxima of the distance function mark the different

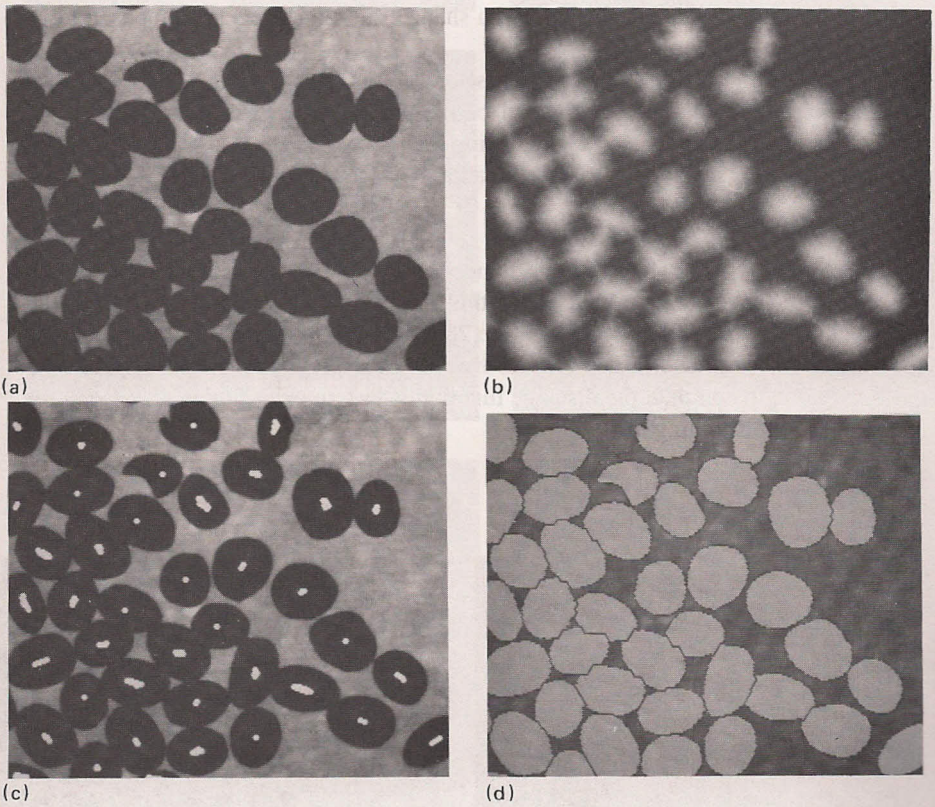


Figure 26. (a) Grains segmentation (coffee beans); (b) distance function; (c) maxima of the distance function; (d) result of the segmentation by thalweg lines of the distance function.

grains (Figure 26c). They can be used (after a slight filtering to solve some parity problems on the digitization grid) to build the talweg lines, defined as the watershed lines of the inverted distance function (Figure 26d).

C. Stereoscopic Analysis of a Fracture in Steel

This third example is a problem of segmentation of cleavage facets in a scanning electron micrograph of a steel fracture (figure 27). The marker selection in this case is more complex. A primary definition of a facet is used: a facet is supposed to be a more or less convex and homogeneous region of the image. That is why the functions used for the watershed along with the marker selection are built by combining a photometric criterion (contrast between facets due to variations in gray tones or to blazing ridges) and a shape criterion (facets are supposed to be more or less convex).

Two functions are defined: the first one, f_1, is the maximum of the gradient function of the initial image f and of the top-hat transformation. The top-hat transform WTH(f) is used for enhancing in the image the blazing zones while the gradient detects the contrast between adjacent facets (Figure 28a):

$$f_1 = \max(g(f), \mathrm{WTH}(f))$$

The second function f_2 is the distance function to the blazing zones and to the contours. It can be shown [5] that this function may be built by dilating the previous function f_1 by a cone (Figure 28b). This technique allows the combination of the two criteria depicted above. Using a gray-tone image instead of a binary one to compute the distance function is just an extension which avoids an arbitrary thresholding of f_1.

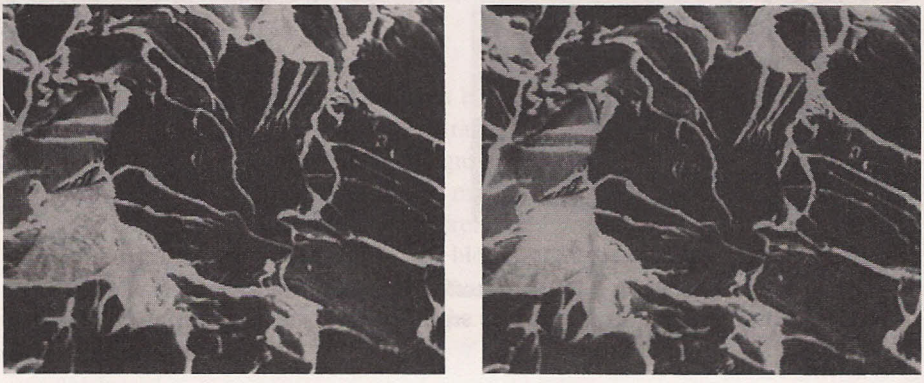


Figure 27. Stereo pair of a cleavage fracture in steel.

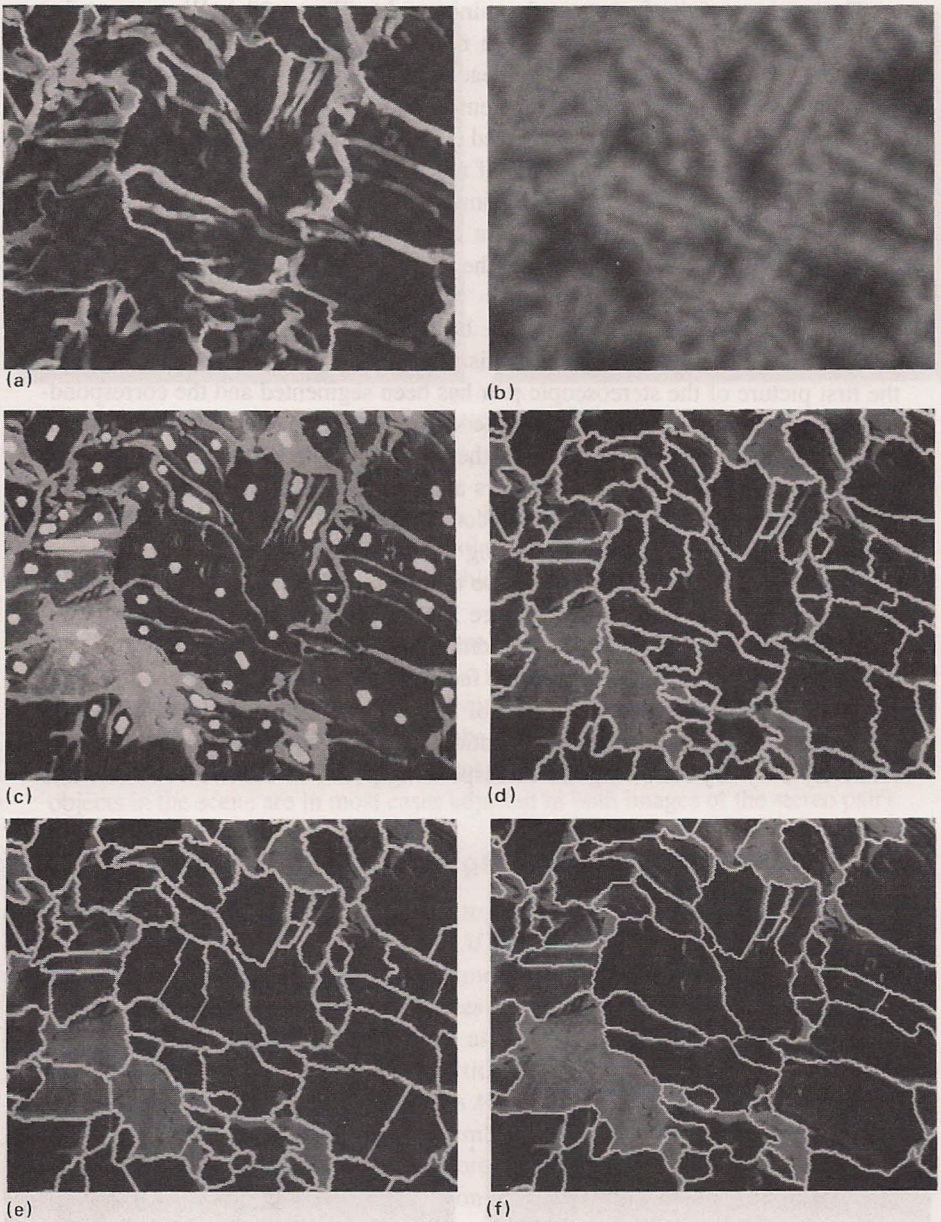


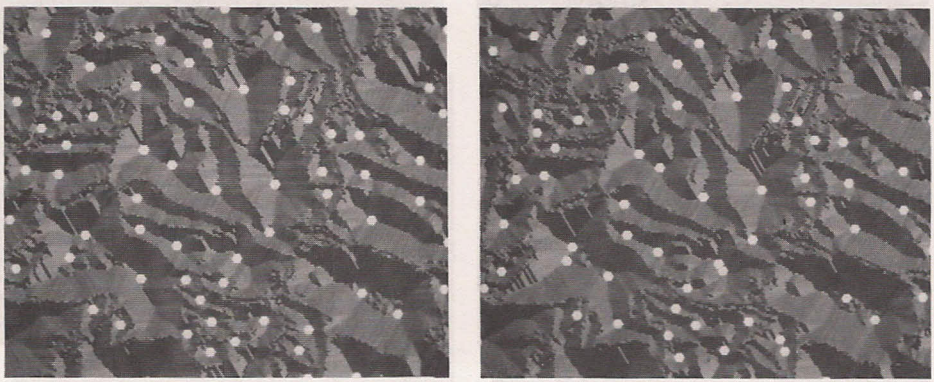
Figure 28. (a) First function used for marker selection; (b) second function; (c) markers of the facets; (d) watershed lines of the first function; (e) watershed lines of the second function; (f) final contours.

The markers of the facets are the minima of f_2 (Figure 28c). We can see that more than one marker may appear in regions which obviously correspond to simple facets. This multiple marking leads to an oversegmentation of the facets.

In order to eliminate this oversegmentation, the watershed transformations of the two functions f_1 and f_2 are performed (Figure 28d and e). These two functions have been modified by the same set of markers (that is, the minima of f_2), and only the divide lines which are superimposed in the two watershed transforms are kept (Figure 28f). This procedure allows one to distinguish between the watershed lines which do not follow the highly contrasted regions in the initial image.

The methodology of segmentation based on the primary definition of the markers of the objects to be extracted is particularly helpful here. Indeed, when the first picture of the stereoscopic pair has been segmented and the corresponding facets have been selected, the markers used in this first step can be used again to segment the homologous facets in the second picture of the stereo pair. The procedure is the following: the markers attached to a facet in the first image are "thrown" onto the second image f_2' corresponding for the second picture to the image f_2. These markers fall along the steepest slope of f_2' and each one reaches a unique minimum of f_2'. These minima are the markers of the homologous facet in the second picture (Figure 29). In this way, we establish a one-to-one correspondence between the markers of the two pictures of the stereo pair and, therefore, between the segmented facets (Figure 30).

As soon as the same facet (or part of a facet) has been segmented in the two pictures of the stereo pair, the computation of its size and orientation in space is relatively easy. By following the corresponding points in the two contours, it is



(a) (b)

Figure 29. (a) Markers of the first image; (b) corresponding markers in the second one.

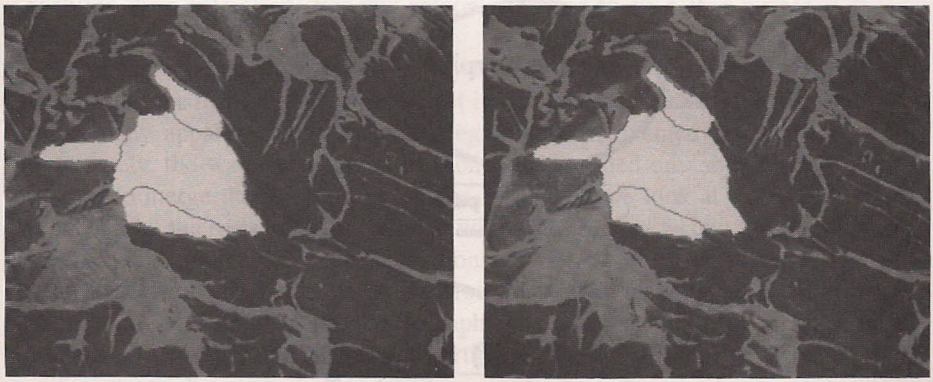


Figure 30. Homologous facets in the stereo pair.

possible to calculate the shift between them and hence their height. Assuming that a facet is almost a plane, its interpolation is performed. Finding the cleavage angle between two adjacent facets (which is in fact the required parameter) is immediate.

This approach to stereovision—segmenting the objects first instead of trying to find the homologous pixels in the two images immediately—is very powerful: the watershed transformation coupled with the marker selection allows us to find directly the corresponding objects in the stereo pair. Moreover, this topological approach allows us to control this correspondence very accurately (two adjacent objects in the scene are in most cases adjacent in both images of the stereo pair).

D. The Segmentation Paradigm

These examples of segmentation lead to a general scheme. Image segmentation consists in selecting first a marker set M pointing out the objects to be extracted and then a function f quantifying a segmentation criterion. This criterion can be, for instance, the changes in gray values, but as seen in the previous examples, other features can be used and even, as illustrated in the case of the cleavage fractures, a mixture of them. This function is modified to produce a new function f' having as minima the set of markers M. The segmentation of the initial image is performed by the watershed transform of f' (Figure 31).

The segmentation process is therefore divided into two steps: an "intelligent" part whose purpose is the determination of M and f and a "straightforward" part consisting in the use of the basic morphological tools, namely the watershed transform and image modification.

This technique has demonstrated its efficiency in various domains of image analysis for both binary and gray-tone pictures. This methodology is also helpful

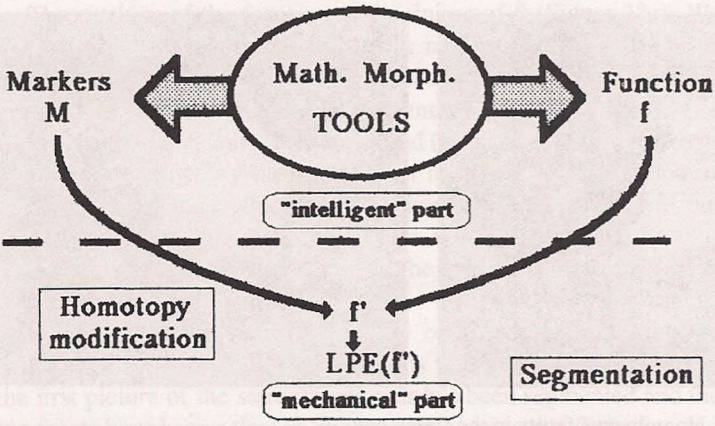


Figure 31. Synopsis of the morphological segmentation methodology.

in three-dimensional segmentation [24], in color image contouring [25], or for the extraction and tracking of objects in time sequences [26].

Many other examples of segmentation based on this scheme may be found in the literature [5,6,27–29].

VII. HIERARCHICAL SEGMENTATION

Until now, our aim was to prevent oversegmentation in selecting good markers and, by means of homotopy modification, to produce as many catchment basins in the watershed as we had selected objects.

This last part will be devoted to the description of a hierarchical segmentation technique which does not prevent oversegmentation but instead tries to suppress the irrelevant boundaries on the watershed transform. We shall see that this approach also uses a watershed transformation defined in this case on a simplified version of the initial image.

A. Introduction

The previous examples have proved that the marker extraction and the good choice of the function used in the watershed are the intelligent process of the segmentation, needing a great deal of effort and skill. The final result is therefore closely dependent on this first task.

Unfortunately, in some cases, marker selection and extraction are not easy. Some pictures are very noisy and image processing becomes more and more complex. In other cases, the objects to be detected may be so complex and so varied in shape, gray level, and size that it is very hard to find reliable algorithms

enabling their extraction. For that reason, we need to go a step further in the segmentation.

When attempting to segment a gray-tone image, we know that the initial watershed transformation of the gradient image provides very unsatisfactory results: many apparently homogeneous regions are fragmented in small pieces. Fortunately, the watershed transformation itself, applied on another level, will help us to merge the fragmented regions. Indeed, if we look at the boundaries produced by the segmentation, they do not have the same weight. Those which are inside the almost homogeneous regions are less significant. In order to compare these boundaries, we need to introduce neighborhood relations between them through the definition of a new graph. This graph is built from a simplified version of the original image called a partition or mosaic image.

B. The Mosaic Image

Although the construction of the mosaic image is not necessary for defining the hierarchical segmentation, it will help for understanding the procedure.

Consider a gray-tone image f and its corresponding morphological gradient image $g(f)$. A simplified image can be computed in the following way:

1. We calculate the watershed of the gradient image.
2. We label every catchment basin of the watershed with the gray value in the initial image f corresponding to the minima of $g(f)$.

Figure 32 illustrates this operation.

We will describe the principle of the hierarchical segmentation by means of a simple example. The initial image is an X-ray photograph of metallic particles in the burst produced by a shaped-charge weapon (Figure 33a).

The result is a simplified image (Figure 33b), made of a mosaic of pieces (the catchment basins) of constant gray levels, where no information regarding the contours has been lost. This simplified image, also called a mosaic image, may then be used to define a valued graph, to which the morphological transforms, and in particular the watershed, can be extended.

C. Hierarchical Segmentation

1. An Introduction

When we look at the mosaic image, some regions seem to be almost homogeneous. In fact, they are made of a patchwork of pieces of constant gray levels, the variation in gray values (the step) between two adjacent tiles being low. On the contrary, when we cross a boundary separating two different regions, the step is much higher. In other words, the criterion used to decide whether we are inside an homogeneous region is the fact that the transitions between the different tiles of the mosaic image which partition this homogeneous region are lower than the

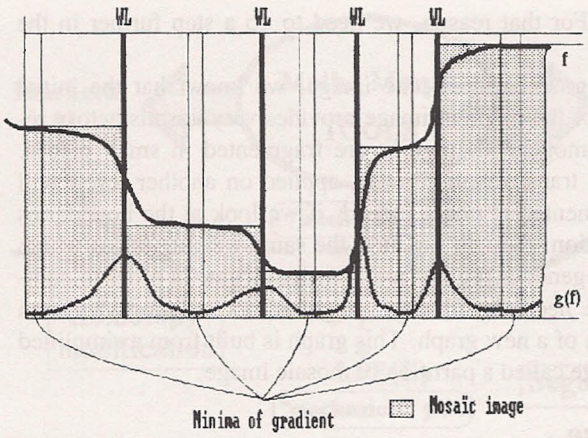
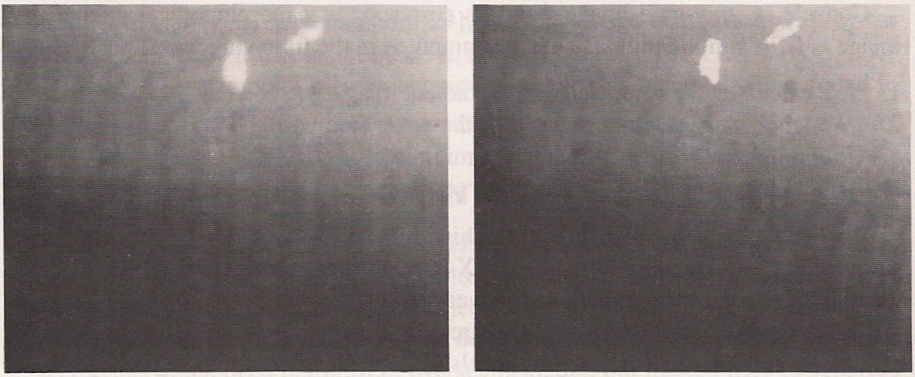


Figure 32. Computation of the mosaic image.



(a) (b)

Figure 33. (a) Initial and (b) mosaic image of an X-ray photograph of metallic particles.

transitions between tiles belonging to different homogeneous regions. Going a step further, one can say that a homogeneous region is marked by minimal transitions in the mosaic picture.

Figure 34a illustrates this notion in a very simple case. The step in gray values between the two tiles supposed to belong to the same homogeneous region is lower than the surrounding ones.

A hierarchical segmentation will then consist in merging adjacent tiles of the original mosaic image using a flooding process starting from the minimal transi-

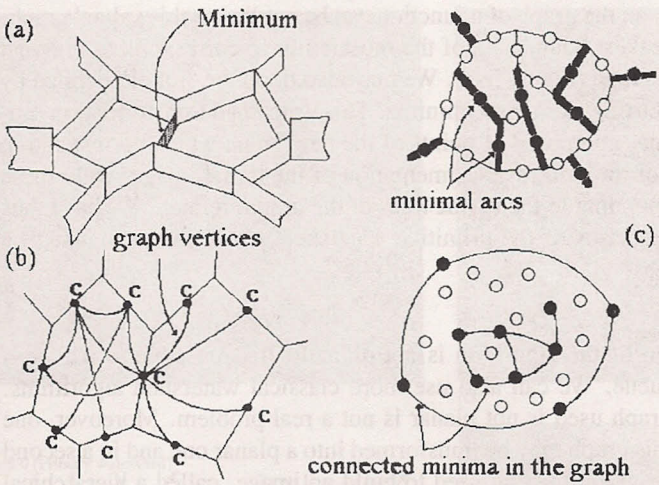


Figure 34. (a) Gradient of the mosaic image; (b) corresponding graph used in the hierarchical approach; (c) example of minimal arcs and their correspondence in the graph.

tions. But this watershed transformation needs, to be realized, the definition of a particular graph built from the mosaic.

2. Construction of a New Graph

Let us build a new valued graph from the mosaic image. First, the summits are made of the transitions between tiles. These summits are valued with the absolute value of the gray-tone difference (the step) between these tiles. Second, the vertices of this new graph, hence the neighborhood relationships between the transitions in the mosaic image, must be set. Two boundaries of the mosaic (two summits of the graph) are considered neighbors if they surround the same catchment basin. This rule simply means that any action on a boundary between two pieces of the mosaic will affect the pieces themselves and therefore the other boundaries which contour them.

The valuation of this graph may be calculated by means of the gradient of the mosaic image (Figure 34a). This morphological gradient is obtained by performing the difference between the dilation and the erosion of the mosaic image. The final graph (Figure 34b) is a nonplanar valued graph.

3. Watershed on the Graph and Hierarchy

All the morphological transformations can be extended to the graph defined above, where the summits correspond to the simple arcs of the primitive watershed transform and the vertices connect the boundaries surrounding the same primitive catchment basin. In particular, the notion of minimum as it has been

defined using paths on the graph of a function can be applied to this valued graph. In our case, the weakest boundaries of the mosaic image correspond to regional minima of the new graph (Figure 34c). We may also flood the "relief" defined by this valued graph starting from these minima. This watershed transformation produces watershed lines composed of points of the new graph which correspond in fact to boundaries of the primitive segmentation of the initial image. Only these boundaries, corresponding to the divide lines of the graph, remain. We have thus suppressed the boundaries of the primitive watershed which are surrounded by more contrasted ones.

4. Illustration

The implementation of the algorithm is not difficult. It is possible to realize it with an ordered queue. We can also use more classical watershed algorithms. The fact that the graph used is not planar is not a real problem. Moreover, one can show [5] that this graph may be transformed into a planar one and in a second step that this planar graph may be used to build an image, called a hierarchical image, to which the classical watershed algorithms may be applied, producing as a result the hierarchical segmentation described above.

Let us show this algorithm on our simple example. To contour the metallic particles, we compute the mosaic image and we suppress some oversegmented regions by performing the first degree of this hierarchical process. But this procedure may be repeated, giving higher and higher levels of hierarchy. At the end of these iterations, the final image is obviously empty. Nevertheless, it is possible to label every boundary of the primitive watershed image with the higher level of hierarchy in which this boundary remains (Figure 35b). It is then easy to see that the particles correspond to the maxima of this new image.

The result of this hierarchical segmentation is given in Figure 35c. From that picture, the extraction of the particles is straightforward. They correspond to the new homogeneous regions that contain the maxima of the initial image (Figure 35d).

D. Another Example

This hierarchical segmentation can be used efficiently for extracting features from complex scenes [30]. Let us apply this technique for delineating the road in the scene represented in Figure 36.

The initial image having been very noisy, the result of the watershed transformation of the gradient image is oversegmented (Figure 36b). However, the road being a rather homogeneous region in the image, we hope that the hierarchical segmentation will help in extracting it from the image. The mosaic image is performed (Figure 36c), and then its gradient (Figure 36d). The result of the first level of hierarchy is given in Figure 36e. In the second step, the region in front of the scene may be extracted. This marker of the road may be used again in an

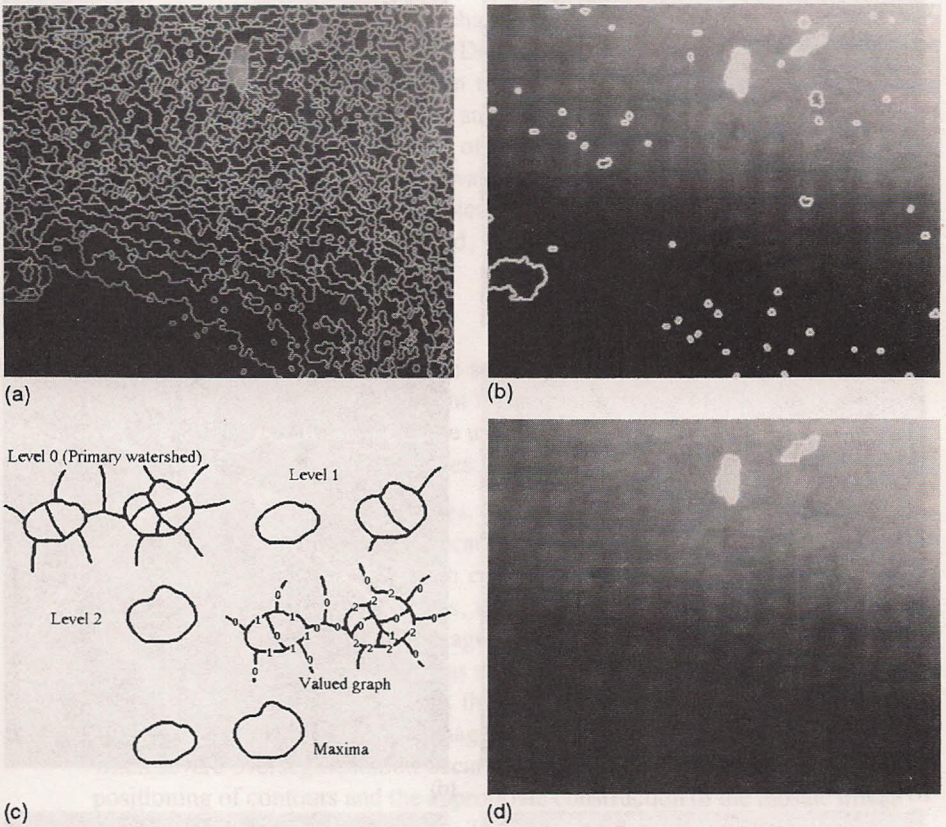


Figure 35. (a) Initial watershed; (b) hierarchical segmentation; (c) principle of labeling of the arcs of the watershed; (d) final result.

homotopy modification and watershed procedure to produce a more refined contouring of the road (Figure 36f).

E. Discussion

The result of the watershed transformation yields to a hierarchical segmentation of the image, as illustrated in the previous examples. The selection of some markers can be made at this level to segment features in the image (for example, the road in the last example). Further levels of hierarchy may also be defined by iterating this procedure (as shown in the introductory example).

Starting from a highly fragmented image, we have obtained after simplification a new mosaic. It is obviously possible to iterate this simplification process. By this means we get a hierarchy of simplification stages, the last always being a

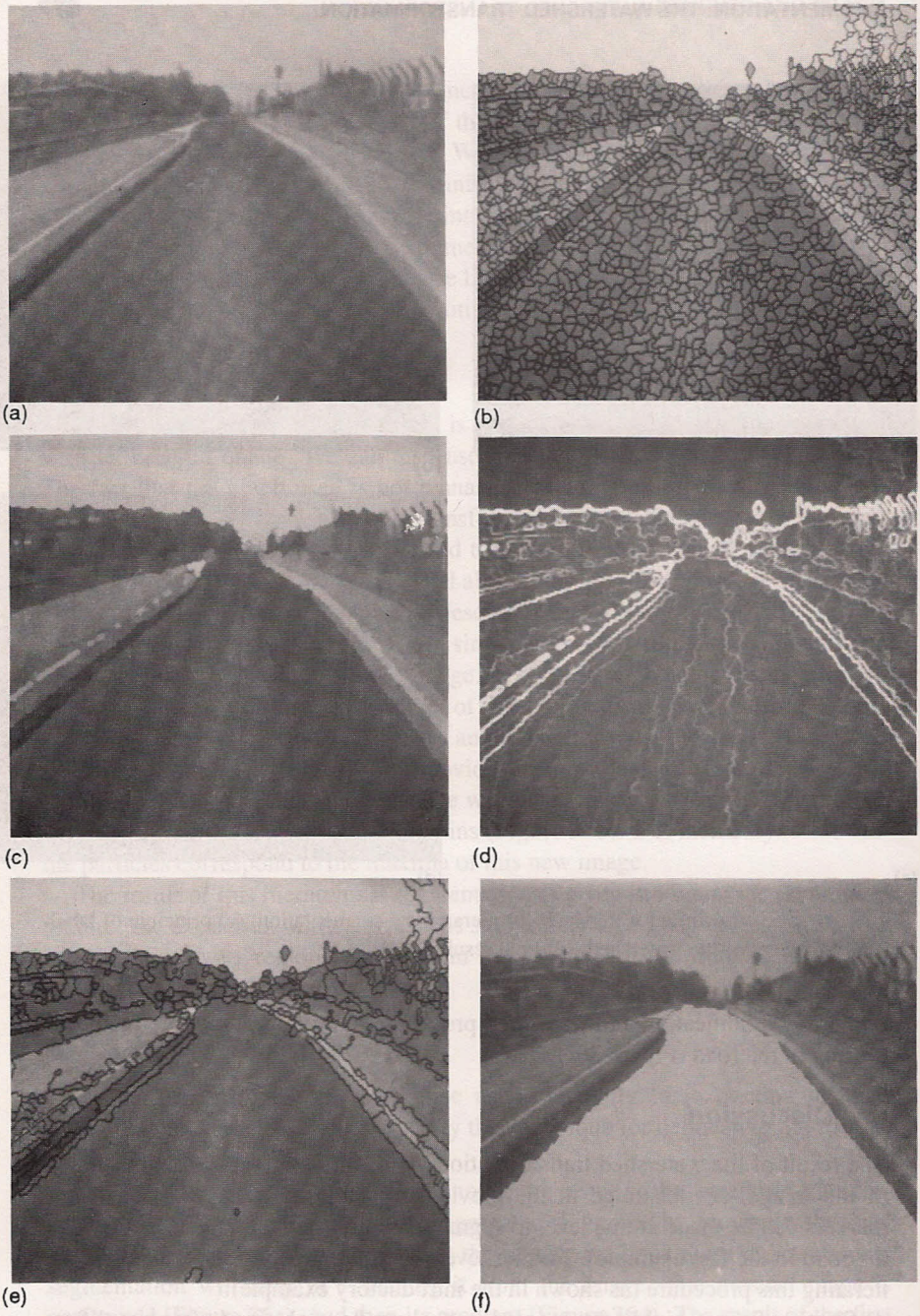


Figure 36. (a) Road scene viewed through the windscreen of a car; (b) result of the watershed applied to the gradient; (c) mosaic image; (d) gradient of the mosaic image; (e) first stage of the hierarchy; (f) extraction of the marker of the road.

uniform image. It is also possible to change the valuation of the graph between the different stages of hierarchization. Doing so, we can introduce various criteria of hierarchization. For instance, in the case of the fracture image (Section VI), one could use as a valuation the angle between two adjacent facets calculated from the measure of the altitude of the boundary points. Many alternative techniques are also possible, such as valuation according to the size, shape, or orientation of the tiles of the mosaic picture and calculation of the new gradient values we get when the tiles are merged.

VIII. CONCLUSION

The morphological approach to image segmentation problems by means of the watershed transformation is an efficient technique, in terms of the results it produces as well as the control kept by the user on every stage of the process. Let us briefly discuss some of these advantages.

1. The watershed transform provides closed contours, by construction. This fact is of primary importance because we do not have to worry about the contour closing of objects, which could be a problem when using contour detection methods. In other words, the watershed transform aims at extracting objects or regions in the image. This property is particularly helpful when there is no visible contour, as shown in the coffee grains example.
2. When computing the watersheds, the watershed lines always correspond to contours which appear in the image as obvious contours of objects, even when severe oversegmentation occurs. This explains, in particular, the good positioning of contours and the appropriate construction of the mosaic image used in the hierarchical approach. This property is very interesting because it gives to the watershed transformation a great advantage compared to the split-and-merge methods, where the first splitting is often a simple regular sectioning of the image leading sometimes to unstable results.
3. It is a general method which can be applied to many situations. We gave some examples of its use in various applications. But, in fact, these examples are only a small selection of the domains in image analysis where this technique has been used efficiently. Remember that this methodology can be applied to three-dimensional images where the contour detection techniques fail because the notion of contour in a three-dimensional image is not easy to define and to handle.
4. The great advantage of this methodology is that it splits the segmentation process into two separate steps. First, we have to detect what we want to extract: it is the marker selection. Then we have to define the criteria which are used to segment the image. These criteria may be photometric (contrast variations), or based on the shape of the objects, or a combination of both. This combination of different criteria is made easier through the use of powerful morphological tools (geodesic transforms, homotopy modification, and so on) as shown in the examples.

This last assertion means that image segmentation cannot be performed accurately and adequately if we do not construct the objects we want to detect. In this approach, the picture segmentation is not the primary step of image understanding. On the contrary, a fair segmentation can be obtained only if we know exactly what we are looking for in the image.

In fact, there is no general method available to achieve this marker detection and object selection. But why should such a general method exist? The everyday practice of image analysis shows, on the contrary, that in many problems you are not able to see the features you want to extract if you don't know a priori what you are looking for. For instance, we saw in the electrophoresis gel example that the blobs may be extracted individually or as heaps. These are two different ways of seeing the same image. That means that you must often build or construct the objects you want to detect. Image segmentation is not the primary step in image understanding; it is its consequence.

REFERENCES

1. Marr, D., *Vision*, Freeman, San Francisco, 1982.
2. Canny, J. F., Finding edges and lines in images, Artificial Intelligence Laboratory, MIT, Cambridge, Massachusetts, TR-720, 1983.
3. Serra, J., *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982.
4. Coster, M., and Chermant, J. L., *Précis d'analyse d'images*, Editions CNRS, France, 1985.
5. Beucher, S., Segmentation d'images et morphologie mathématique, Doctorate thesis, School of Mines, Paris, 1990.
6. Meyer, F., and Beucher, S., Morphological segmentation. *J. Visual Commun. Image Repres.*, *1(1)*, 21–45 (1990).
7. Meyer, F., Cytologie quantitative et morphologie mathématique, Doctorate thesis, School of Mines, Paris, 1979.
8. Matheron, G., Filters and lattices, in *Image Analysis and Mathematical Morphology*, vol. 2, Academic Press, London, 1988, pp. 115–136.
9. Lantuejoul, C., and Beucher, S., On the use of the geodesic metric in image analysis, *J. Microsc.*, *121*(Pt 1) (1981).
10. Matheron, G., *Random Sets and Integral Geometry*, Wiley, New York, 1975.
11. Meyer, F., Skeletons and perceptual graphs, *Signal Process.*, *16(4)*, 335–363 (1989).
12. Matheron, G., Examples of topological properties of skeletons, in *Image Analysis and Mathematical Morphology*, vol. 2, Academic Press, London, 1988, pp. 217–233.
13. Beucher, S., and Lantuejoul, C., Use of watersheds in contour detection, in *Proceedings, International Workshop on Image Processing, CCETT/IRISA*, Rennes, France, 1979.
14. Serra, J., *Image Analysis and Mathematical Morphology*, vol. 2, *Theoretical Advances*, Academic Press, London, 1988.

15. Meyer, F., Sequential algorithms for cell segmentation: maximum efficiency? in *Proceedings, International Symposium on Clinical Cytometry and Histometry*, Schloss Elmau, 1986.
16. Rosenfeld, A., and Pfaltz, J. L., Sequential operations in digital picture processing, *J. ACM*, *13*, 471–494 (1966).
17. Borgefors, G., Distance transformations in digital spaces, *CVGIP*, *34*, 344–371 (1986).
18. Danielson, P. E., Euclidean distance mapping, *CVGIP*, *14*, 227–248 (1980).
19. Meyer, F., Algorithmes séquentiels, *Proceedings, Onzième Colloque GRETSI*, Nice, France, 1987, pp. 543–546.
20. Vincent, L., Algorithmes morphologiques à base de files d'attente et de lacets. Extension aux graphes, Doctorate thesis, School of Mines, Paris, 1990.
21. Verwer, B. J. H., Verbeek, P. W., and Dekker, S. T., An efficient uniform cost algorithm applied to distance transforms, *PAMI*, *11*, 425–429 (1989).
22. Verbeek, P. W., and Verwer, J. H., Shading from shape, the eikonal equation solved by grey-weighted distance transformation, *Pattern Recogn. Lett.*, *11*, 681–690 (1990).
23. Meyer, F., Un algorithme optimal de partage des eaux, in *Proceedings 8th Congress AFCET*, Lyon-Villeurbanne, France, 1992, vol. 2, pp. 847–859.
24. Gratin, C., and Meyer, F., Mathematical morphology in three dimensions, in *Proceedings 8th ICS*, Irvine, California, Aug. 25–30, 1991.
25. Meyer, F., Color image segmentation, in *Fourth International Conference on Image Processing and Its Applications*, Maastricht, April 1992.
26. Friedlander, F., Le traitement morphologique d'images de cardiologie nucléaire, Doctorate Thesis, School of Mines, Paris, 1989.
27. Beucher, S., Segmentation tools in mathematical morphology, in *Proceedings, SPIE Congress*, San Diego, 1990.
28. Vincent, L., and Soille, P., Watersheds in digital spaces: an efficient algorithm based on immersion simulations, *IEEE PAMI*, *1*(6), 583–597 (1990).
29. Beucher, S., The watershed transformation applied to image segmentation, in *Tenth Pfefferkorn Conference*, Cambridge, UK, Scanning Microscopy International, 1991.
30. Beucher, S., Bilodeau, M., and Yu, X., Road segmentation by watershed algorithms, in *Proceedings, Prometheus Workshop*, Sophia-Antipolis, France (1990).