
Current Topics in Applied Morphological Image Analysis

Luc Vincent

To appear in *Current Trends in Stochastic Geometry and its Applications*, W.S. Kendall, O.E. Barndorff-Nielsen, M.C. Van Lieshout, editors, Chapman & Hall, London, 1997.

1.1 Introduction

Morphology is an image processing methodology which was born in France in the late sixties and has been gaining increasing importance and popularity since then (Matheron 1975, Serra 1982, Serra 1988, Schmitt and Vincent 1997). Today, morphology is broadly accepted as one of the most useful “toolboxes” for a wide variety of image analysis applications. A solid knowledge of this methodology enables the quick development of algorithms and products for solving complex detection, segmentation, and pattern recognition problems. Morphology has been successfully applied in many areas, including medicine (MR imagery, CT imagery, X-rays, angiographies, etc), biology, radar, sonar, infra-red, and remote sensing images, industrial inspection, robotics, material science, fingerprints, identification, document recognition, etc (See Fig. 1.1). It addresses binary and grayscale images, as well as color and 3-D imagery. A priori, any application involving images sharing a well-defined set of characteristics (e.g., size and shape of the objects to be detected in them) is a good candidate for a morphological solution.

The morphological approach to image analysis is natural and attractive: it begins by considering binary images as sets and grayscale images as functions or topographic reliefs. These sets and functions are then transformed—in the spatial domain—via morphological operators, whose definitions are usually based on *structuring elements*, i.e. particular shapes that are translated in the images and used as probes. From these basic operations, in-

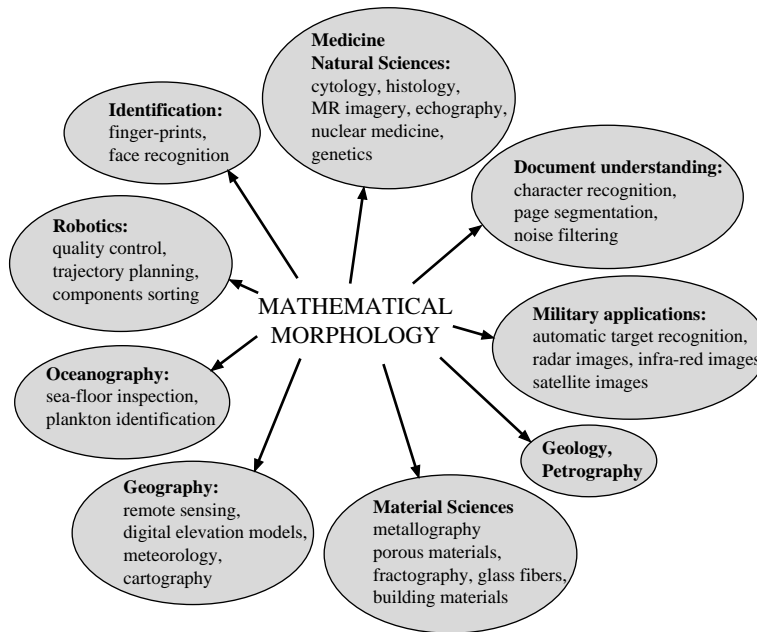


Figure 1.1 *Fields of application of morphological image analysis.*

creasingly complex operators are derived, that are used towards more and more specific goals, like the detection of gaps and protrusions, valley and crest-line extraction in grayscale images, shape- and size-based feature extraction, etc. Despite this increasing algorithmic complexity, the operators remain extremely easy to use: they are intuitive, flexible, and powerful.

In the present chapter, we focus on two very very important aspects of mathematical morphology: image segmentation and granulometries. Segmentation is the task of partitioning an image into its “meaningful” regions. This is often equivalent to detecting and extracting from an image the objects or zones of interest. These objects or zones can then be counted, measured, classified, etc. Accurate segmentation is therefore a critical step in a number of image analysis systems. On the other hand, granulometries are concerned with extracting size information from images *without* prior segmentation. This can be key when either the images under study are too difficult or too intricate to be segmented, or when process-

ing speed is critical, or when size information is itself needed to guide the segmentation process. Beyond this, granulometries provide a set of general purpose feature extractors that turn out to be very powerful in a number of classification applications. Both morphological segmentation techniques and granulometries have seen significant advances in the past decade, and hundreds of papers have been published on these topics.

Our approach here is pragmatic and applied: indeed, in the field of image processing, the intuitive understanding of a few key concepts is usually more important than the knowledge of their theoretical foundations. Besides, a single chapter would not be sufficient to thoroughly cover both the theoretical and the applied aspects of segmentation and granulometries. For mathematical morphology theory, we invite the reader to refer to the monograph by Matheron (1975), the classic books by Serra (1982, 1988), or the more recent monograph by Heijmans (1994). French speaking readers can also consult the excellent new book by Schmitt and Mattioli (1994). Finally, it is worth mentioning that the topics covered in the present chapter will also be extensively discussed in a book to be published in 1997 (Schmitt and Vincent 1997).

This chapter is organized as follows: we begin in Section 1.2 with a brief review of mathematical morphology, introduce a few notations and conventions, and provide some reminders on morphological concepts that will be useful throughout the chapter. In Section 1.3, the emphasis is put on morphological segmentation, and such tools as *grayscale reconstruction*, *area openings*, and *watersheds* are described in detail. The effects of these operators on images are illustrated on numerous examples, and their use for image analysis problem-solving is emphasized. Finally, Section 1.4 deals with granulometries. In this section, while a number of applications are also presented, more emphasis is put on the *algorithmic* aspect. Indeed, regardless of how powerful a morphological operator is “on paper”, if no efficient implementation exists, this operator does not serve any practical purpose. Efficient binary and grayscale granulometry algorithms are described, and their use is illustrated in a variety of applications, ranging from size estimation, texture characterization, and classification. Problem-solving and algorithms are the two aspects of applied morphology that this chapter is mostly concerned with.

1.2 Brief Review of Mathematical Morphology

For the sake of completeness of this chapter, the present section provides some background on mathematical morphology, its classic operators, and associated notations. Readers who are already very familiar with such concepts may skip directly to Section. 1.3; others will find here some reference material that will greatly help them understand Sections 1.3 and 1.4.

1.2.1 Basic Definitions and Notations

Definition 1 *A two-dimensional binary image is a subset X of the continuous plane \mathbb{R}^2 .*

In the following, we exclusively deal with *bounded* sets. Besides, X is often considered equivalent to its *characteristic function*, i.e., the mapping f from \mathbb{R}^2 into $\{0, 1\}$ such that $f(p) = 1$ if $p \in X$ and $f(p) = 0$ otherwise.

Definition 2 *A two-dimensional grayscale image I is defined as a mapping from a bounded subset D_I of \mathbb{R}^2 (the domain of I) into \mathbb{R} :*

$$p \in D_I \longrightarrow I(p) \in \mathbb{R}.$$

Grayscale images are often called *functions* in the morphological literature. In the following, all the grayscale images we deal with are defined on a rectangular domain. From the two above definitions, it is clear that binary images are simply grayscale images that only take values 0 and 1. Therefore, any morphological operator defined for grayscale images is a perfectly valid binary operator as well.

The *operators* or *transformations* ψ we deal with typically transform binary images (i.e. sets) into binary images, or grayscale images into grayscale images. Some operator like the *threshold operation* transform a grayscale image into a binary image. Similarly, operators like the *distance function* turn a binary image into a grayscale image. Note that morphological operators can be defined in more general spaces such as *complete Boolean lattices* (Birkhoff 1983). Such theoretical aspects of morphology would however be beyond the scope of this chapter (See Heijmans 1994 for more detail).

Now, let ψ be a binary operator, i.e., a mapping from $\mathcal{P}(\mathbb{R}^2)$ into $\mathcal{P}(\mathbb{R}^2)$:

$$X \in \mathbb{R}^2 \longmapsto \psi(X) \in \mathbb{R}^2.$$

The following definitions will be useful throughout the present chapter:

Definition 3 ψ is said to be an extensive transformation if and only if, for any set X , $X \subseteq \psi(X)$. Similarly, ψ is said to be anti-extensive if and only if, for any set X , $\psi(X) \subseteq X$.

Definition 4 ψ is increasing if and only if it preserves the inclusion relations between sets, i.e.:

$$X \subseteq Y \implies \psi(X) \subseteq \psi(Y). \quad (1.1)$$

Definition 5 ψ is said to be idempotent when applying it several times in a row is equivalent to applying it only once:

$$\text{For any set } X, \quad \psi(\psi(X)) = \psi(X). \quad (1.2)$$

Finally, with ψ' being another binary operator, we can give the following definition:

Definition 6 ψ and ψ' are said to be dual of each other if applying one to a set X is equivalent to applying the other to the complement X^C of set X :

$$\psi(X)^C = \psi'(X^C). \quad (1.3)$$

In the previous definition, the complement operator is denoted by superscript C : $X^C = \mathbb{R}^2 \setminus X$.

To extend the above definitions to grayscale images, we need to define a partial order between functions. The following is used:

$$f \geq g \iff \forall x, f(x) \geq g(x). \quad (1.4)$$

Similarly, the ‘‘complement’’ of a function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ is taken to be the function $(-f)$. Given this, definitions 3 to 6 easily extend to grayscale.

Most of the concepts this chapter deals with also extend to multidimensional images, or can be restricted to 1D signals. However, for simplicity, only the 2D case is considered here. Furthermore, although many of the concepts and methods discussed extend to color and multispectral images, due to the limitations of a single chapter, we must leave such topics aside.

In practice, the images we deal with are *discrete*, i.e., defined on a (rectangular) subset of the discrete plane \mathbb{Z}^2 . Binary images take values 0 and 1 (the pixels with value 1 are sometimes called the ON pixels) whereas grayscale images take their values in the range $\{0, 1, \dots, N\}$. For most applications, 8-bit per pixel is sufficient,

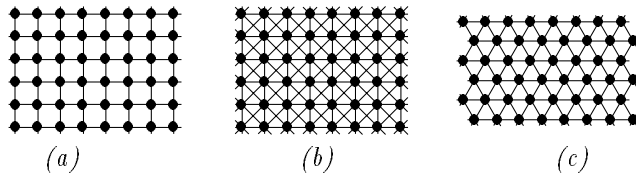


Figure 1.2 *The three classic grids used in discrete images. (a) 4-connected square grid; (b) 8-connected square grid; (c) hexagonal grid (c-connectivity).*

therefore $N = 255$. In the following, new concepts are often defined in the continuous plane, and then applied to discrete images: indeed, the discrete version of a morphological operator is usually easy to derive from the “continuous definition”. When it is more convenient to our purpose, we define the operator directly in the discrete plane \mathbb{Z}^2 .

In the discrete plane, a *grid* G provides the connectivity, i.e., the neighborhood relationships between pixels. Commonly used grids are the square grid, for which a pixel p has either 4 (in 4-connectivity) or 8 neighbors (in 8-connectivity), as well as the hexagonal grid (6-connectivity). This is illustrated by Fig. 1.2. Two neighboring pixels p and q form an edge of G . The grid G induces a discrete distance in \mathbb{Z}^2 , the distance between two pixels being the minimal number of edges required to join them. This distance metrics is denoted here by d_G , and in 4-, 6-, and 8-connectivity, we simply use d_4 , d_6 , and d_8 . In a grid G , we denote by $N_G(p)$ the set of neighbors of a given pixel p . In 4-, 6- and 8-connectivity, we sometimes denote $N_G(p)$ by $N_4(p)$, $N_6(p)$ and $N_8(p)$ respectively.

In all our examples, binary images are printed “black on white”, that is, the ON pixels (pixels with value 1) are shown in black on a white background corresponding to the OFF pixels. The opposite convention is used for grayscale images: pixel values correspond to their brightness level. See Fig. 1.3 for examples of binary and grayscale images.

1.2.2 Basic Morphological Operators, Binary Case

As mentioned earlier, the basic morphological operators are defined based on the concept of structuring element. A structuring element is a particular set $B \subset \mathbb{R}^2$, usually small and of simple shape (disk,

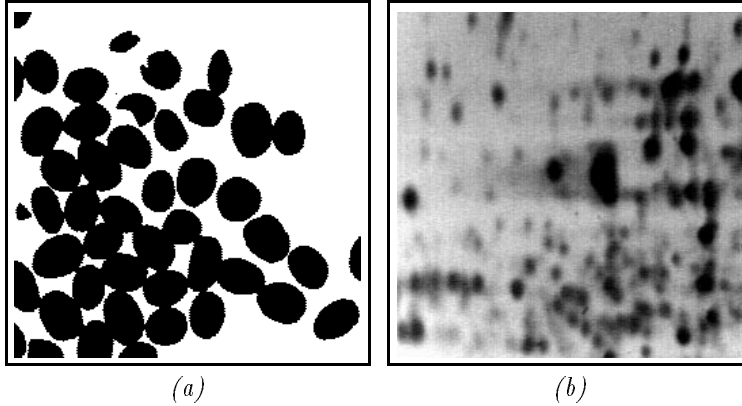


Figure 1.3 (a) Binary image of coffee beans: the ON pixels (value 1) are shown in black; (b) grayscale image of two-dimensional electrophoresis gel: dark spots correspond to low pixel values.

square, line segment, etc) that gets translated over an image X and whose relationships with this image are studied at each location. We denote by \check{B} the *transposed set*:

$$\check{B} = \{-b \mid b \in B\}. \quad (1.5)$$

\check{B} is simply obtained by rotating B by 180° about the origin O , which typically belongs to B . (The origin O is often called the *center* or *hot point* of structuring element B .) We also denote by B_x the translation of B by x :

$$B_x = \{b + x \mid b \in B\}. \quad (1.6)$$

Definition 7 The dilation of a set $X \subset \mathbb{R}^2$ by a structuring element B , denoted $\delta_B(X)$, is the set of points $x \in \mathbb{R}^2$ such that the translation of B by x has a non-empty intersection with set X :

$$\delta_B(X) = \{x \in \mathbb{R}^2 \mid X \cap B_x \neq \emptyset\}. \quad (1.7)$$

One can show that the dilation of X by B , is equal to the *Minkowski Addition* (Minkowski 1897) of X and \check{B} , denoted $X \oplus \check{B}$:

$$\delta_B(X) = X \oplus \check{B} = \{x + b \mid x \in X, b \in \check{B}\}. \quad (1.8)$$

Note that some authors actually define the morphological dilation as a Minkowski addition (with no transposition of the structuring element). If we grossly generalize, we can say that the French

school (Serra 1982) uses definition 7 whereas the American school (Sternberg 1986) defines dilation to be equal to the Minkowski addition. In practice, such distinctions are mostly academic, since useful structuring elements B are typically symmetric: $\check{B} = B$.

Definition 8 *The erosion of X by structuring element B , denoted $\varepsilon_B(X)$, is the set of points $x \in \mathbb{R}^2$ such that the translation of B by x is included in X :*

$$\varepsilon_B(X) = \{x \in \mathbb{R}^2 \mid B_x \subseteq X\}. \quad (1.9)$$

Here again, this definition of the erosion of X by B corresponds to the *Minkowski subtraction* of X and \check{B} :

$$\varepsilon_B(X) = X \ominus \check{B} = \{x \in \mathbb{R}^2 \mid \forall b \in \check{B}, x - b \in X\}. \quad (1.10)$$

An example of erosion and dilation is shown in Fig. 1.4. From this illustration, it is clear that the dilation operator tends to “grow” sets while the erosion “shrinks” them. Through dilation, one can connect different components of a set X , fill some holes and gaps, whereas through erosion, holes become bigger, and objects (i.e., connected components) can vanish. More formally, these operations are equipped with a number of fundamental properties:

1. Erosion and dilation are *increasing* operators:

$$X \subseteq Y \implies \varepsilon_B(X) \subseteq \varepsilon_B(Y) \text{ and } \delta_B(X) \subseteq \delta_B(Y). \quad (1.11)$$

2. Erosion and dilation are *dual* operations:

$$\varepsilon_B(X) = [\delta_B(X^C)]^C \text{ and } \delta_B(X) = [\varepsilon_B(X^C)]^C. \quad (1.12)$$

3. When the structuring element “contains its center”, i.e., when $\vec{o} \in B$, then the erosion by B is anti-extensive and the dilation by B is extensive:

$$\vec{o} \in B \implies \varepsilon_B(X) \subseteq X \text{ and } X \subseteq \delta_B(X). \quad (1.13)$$

4. Commutativity with union/intersection:

$$\delta_{B_1 \oplus B_2}(X) = \delta_{B_1}(\delta_{B_2}(X)) = \delta_{B_2}(\delta_{B_1}(X)) \quad (1.14)$$

$$\varepsilon_{B_1 \oplus B_2}(X) = \varepsilon_{B_1}(\varepsilon_{B_2}(X)) = \varepsilon_{B_2}(\varepsilon_{B_1}(X)). \quad (1.15)$$

5. Structuring element combinations:

$$\delta_{B_1 \oplus B_2}(X) = \delta_{B_1}(\delta_{B_2}(X)) = \delta_{B_2}(\delta_{B_1}(X)) \quad (1.16)$$

$$\varepsilon_{B_1 \oplus B_2}(X) = \varepsilon_{B_1}(\varepsilon_{B_2}(X)) = \varepsilon_{B_2}(\varepsilon_{B_1}(X)) \quad (1.17)$$

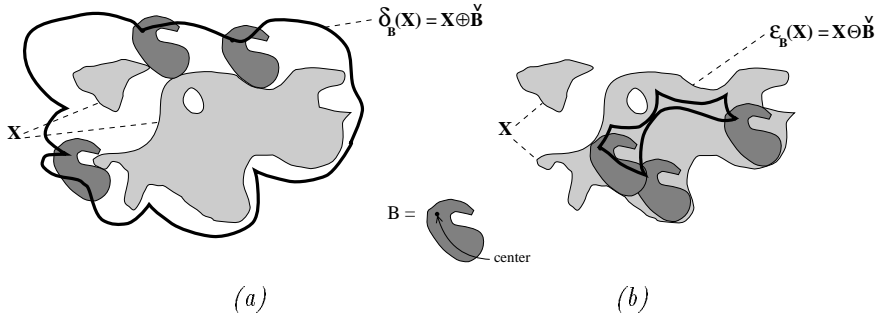


Figure 1.4 (a) Dilation of set X by structuring element B . (b) Erosion of X by B . In each case, thick black outline represents contour of resulting set.

$$\delta_{B_1 \cup B_2}(X) = \delta_{B_1}(X) \cup \delta_{B_2}(X) \quad (1.18)$$

$$\varepsilon_{B_1 \cup B_2}(X) = \varepsilon_{B_1}(X) \cap \varepsilon_{B_2}(X) \quad (1.19)$$

Property number 2 above is particularly interesting in practice: it means that eroding the foreground is equivalent to dilating the background. Stated differently, given a dilation operator, we can obtain the corresponding erosion by complementing, dilating, and complementing again. The same holds for a number of pairs of basic morphological operators.

Property 5 is also of practical interest: given a complex structuring element B , if we can find a decomposition of B as a union and/or Minkowski addition of simpler elements, then we can perform a dilation (resp. erosion) by B as a combination of dilations (resp. erosions) by simpler shapes. This can be extremely valuable from an algorithmic point of view. A specific instance of this property concerns the *homothetics* (i.e., scaled versions) of a basic structuring element. Denote by nB the n -fold Minkowski addition of B to itself:

$$nB = \underbrace{B \oplus B \oplus \dots \oplus B}_{n \text{ times}}, \quad (1.20)$$

with $1B = B$. Then dilating a set X by nB is equivalent to iterating n dilations of X by B :

$$\delta_{nB}(X) = \underbrace{\delta_B(\delta_B(\dots \delta_B(X)))}_{n \text{ times}}. \quad (1.21)$$

Figure 1.5 *Unit-size ball B for different grids*

Similarly for erosions:

$$\varepsilon_{nB}(X) = \underbrace{\varepsilon_B(\varepsilon_B(\dots\varepsilon_B(X)))}_{n \text{ times}}. \quad (1.22)$$

In practice, the most commonly used structuring elements are the homothetics of the *unit-size ball* B for the grid G being used. The unit-size ball centered at pixel p is equal to the union of p and its neighboring pixels for the grid. In other words, it is equal to the pixels (points of \mathbb{Z}^2) whose distance d_G to p is smaller or equal to 1. In 4-, 8-, and 6-connectivity, the unit-size ball B is respectively denoted by D (“diamond”), S (square), and H (hexagon), as illustrated by Fig. 1.5. In a given grid, a dilation (resp. erosion) by the unit-size ball B is often referred to as dilation (resp. erosion) of size 1. Similarly, a dilation (resp. erosion) by nB , which is equal to n consecutive dilations (resp. erosions) by B , is called dilation (resp. erosion) of size n . This loose terminology will be used throughout the chapter.

From the erosion and dilation operators, two new extremely useful morphological transformations can be derived as follows:

Definition 9 *The opening of X by B , denoted by $\gamma_B(X)$, is given by:*

$$\gamma_B(X) = \delta_{\bar{B}}(\varepsilon_B(X)) \quad (1.23)$$

The opening of X by B is often denoted by $(X)_B$, or $X \circ B$, notations that will be used in the present chapter when deemed more convenient.

Definition 10 *The closing of X by B , denoted by $\phi_B(X)$, is given by:*

$$\phi_B(X) = \varepsilon_{\bar{B}}(\delta_B(X)) \quad (1.24)$$

In literature, the closing of X by B is sometimes denoted by $(X)^B$, or $X \odot B$.

Fig. 1.6 illustrates the effect of an opening and a closing by a disk. Clearly, both operators tend to smooth the boundary of the

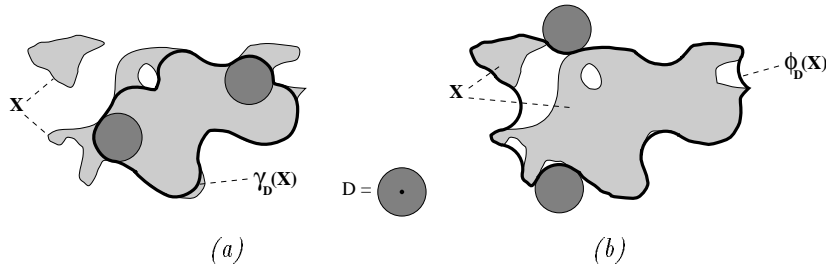


Figure 1.6 (a) Opening of set X by disk D . (b) Closing of X by D . In each case, thick black outline represents contour of resulting set.

original set X , but each in a different manner: the opening tends to remove the small protrusions in which the structuring element cannot fit, whereas the closing fills in the gaps that cannot hold the structuring element. The opening also shrinks the original set whereas the closing makes it bigger. More formally, one can state the following properties:

1. Openings and closings are increasing operators.
2. Openings are anti-extensive, closings are extensive.
3. Openings and closings are duals of each other.
4. Openings and closings are *idempotent*.
5. The opening of X by B is the set of points of X that can be swept by structuring element B when B is translated inside X :

$$\gamma_B(X) = \bigcup_{B_x \subseteq X} B_x. \quad (1.25)$$

A similar property holds for the closing:

$$\phi_B(X) = \left(\bigcup_{B_x \cap X = \emptyset} B_x \right)^C. \quad (1.26)$$

While property 5 above provides an intuitive understanding of openings and closings, properties 1 through 4 are particularly important. For example, properties 1 and 4 together specify that openings and closings are particular types of *morphological filters* (Serra 1988, Serra and Vincent 1992). The theory of morphological filters is extensive and goes beyond the scope of this chapter. However, let us recall two important definitions:

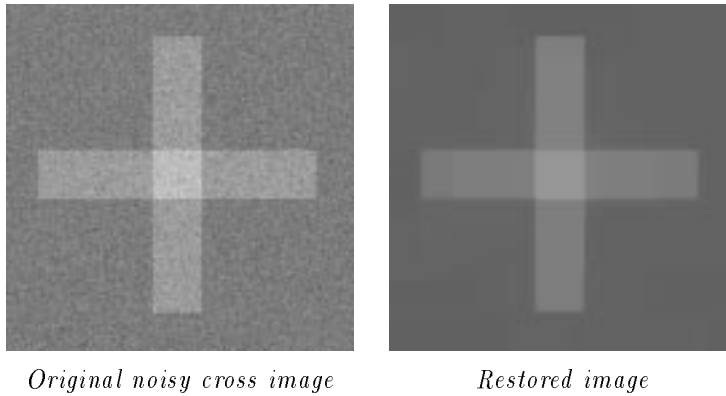


Figure 1.7 *Image restoration using Alternating Sequential Filters.*

Definition 11 A Morphological Filter is an operator that is both increasing and idempotent.

Definition 12 An algebraic opening is an anti-extensive morphological filter, and an algebraic closing is an extensive morphological filter.

Some of the morphological operators we deal with later in this chapter are *algebraic* openings according to Definition 12, but since they cannot be obtained by combining a dilation and an erosion, they are not *morphological* openings.

One can safely state that openings and closing are among the most useful operations in the morphological toolbox. They are directly useful for simple filtering tasks, and by combining them, one can design powerful filters that are tailored to specific applications. For example, Fig. 1.7 illustrates how *Alternating Sequential Filters (ASF)* based on grayscale openings and closings of increasing size (see Section 1.2.3) can be used for the restoration of noisy images (Sternberg 1986, Serra 1998, Serra and Vincent 1992). Furthermore, openings, closings, and the derived *top hat transformation* provide simple and powerful segmentation tools, as described in Section 1.2.3. Finally, openings and closings are at the foundation of *granulometries*, which are discussed in Section 1.4.

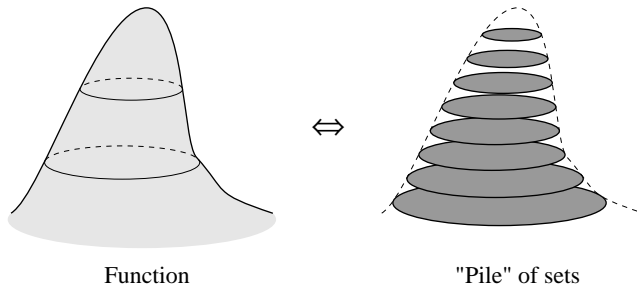


Figure 1.8 *Threshold decomposition of a graylevel image.*

1.2.3 Extension to Grayscale

All four operators described so far are *increasing*, and can therefore be extended to grayscale as follows: let I be a grayscale image, and denote by T_h the operator of *threshold at level h* :

$$T_h(I) = \{p \in D_I \mid I(p) \geq h\}. \quad (1.27)$$

Proposition 13 (Threshold Decomposition) *Let ψ be an increasing binary morphological operator, i.e., for all $X \subseteq Y$, we have $\psi(X) \subseteq \psi(Y)$. ψ can be extended to grayscale as follows:*

$$\psi(I)(p) = \sup\{h \mid p \in \psi(T_h(I))\}, \quad (1.28)$$

where I is a grayscale image and $p \in D_I$.

Indeed, the sets $(T_h(I))$ form a sequence of sets that characterize grayscale image I entirely, and are such that for any thresholds h and h' ,

$$h \geq h' \implies T_h(I) \subseteq T_{h'}(I),$$

as illustrated by Fig. 1.8. By definition, these inclusion relationships are preserved when applying increasing operator ψ , thus creating transformed grayscale image $\psi(I)$. For a more in-depth discussion, refer to .

In particular, using the above proposition, one can easily extend erosions, dilations, openings and closings, to the grayscale case. Intuitively, the above proposition, often referred to as *threshold decomposition* or *threshold superposition* property (Maragos and Ziff 1990), states that to apply an increasing binary operator ψ to a grayscale image I , one can simply apply ψ to every “slice” of I . When this principle is applied to erosions and dilations, one can prove the following:

Proposition 14 *The dilation and erosion of a grayscale image I with respect to structuring element B are given at pixel $x \in D_I$ by:*

$$\delta_B(I)(x) = \sup\{I(p) \mid p \in B_x\} \quad (1.29)$$

$$\varepsilon_B(I)(x) = \inf\{I(p) \mid p \in B_x\} \quad (1.30)$$

This classic result, which can be easily proved (See for example), states that grayscale erosions and dilations can respectively be obtained by computing minima and maxima over a moving window—the structuring element. One can also prove that grayscale openings and closings are simply obtained as combinations of grayscale erosions and dilations:

$$\gamma_B(I) = \delta_{\bar{B}}(\varepsilon_B(I)) \quad \text{and} \quad \phi_B(I) = \varepsilon_{\bar{B}}(\delta_B(I)), \quad (1.31)$$

where I is a grayscale image.

In fact, the dilations (resp. erosions, openings and closings) defined above are often referred to as *flat* dilations, or dilations with *flat* structuring elements. More general dilations can be defined by considering the *umbra* or *subgraph* of grayscale image I :

$$U(I) = \{(x, h) \in \mathbb{R}^2 \times \mathbb{R} \mid h \leq I(x)\}, \quad (1.32)$$

and dilating this three-dimensional set by three-dimensional structuring elements using equation (1.7). The same holds for erosions, openings, and closings, but in general, it has little practical value. In the sequel, we focus on “flat” structuring elements.

Figure 1.9 illustrates the concepts of grayscale dilation and grayscale erosion. In this figure, a cross-section of the original grayscale image (or “function” f) is shown: the horizontal axis corresponds to this cross-section whereas the vertical axis corresponds to grayscale values. On this type of figure, only a cross section of the (flat) structuring element can be shown, i.e., a horizontal line-segment. The (cross-section of the) resulting transformed functions are shown with a thick black stroke. One can observe that the dilation “widens the peaks” and “shrinks the valleys” whereas the erosion has the opposite effect.

Similarly, Fig. 1.10 illustrates the concepts of grayscale openings and closings. Here, the opening tends to clip the narrow peaks in which the structuring element cannot fit, whereas the closing fills-in the valleys that are too narrow for the structuring element.

Aside from a vast variety of morphological filters, a very useful basic morphological segmentation operator can be derived from gray openings and closings: the *top hat* transformation. Originally

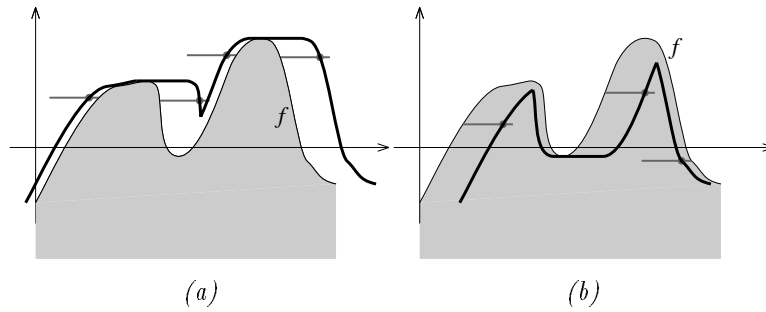


Figure 1.9 Grayscale dilation (a) and erosion (b) of a function f (grayscale image).

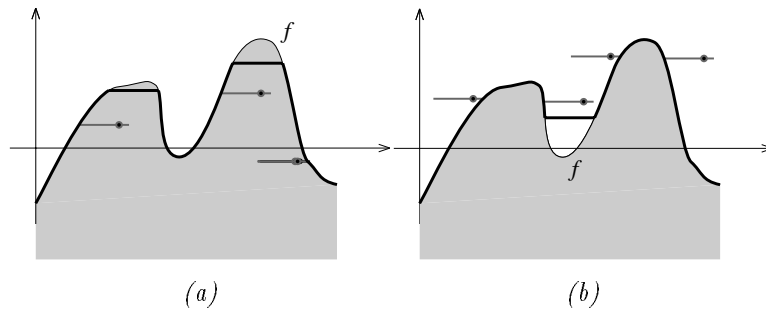


Figure 1.10 (a) Grayscale opening. (b) Grayscale closing.

proposed by Meyer (1978), it provides an excellent tool for extracting light (resp. dark) objects from an uneven background. It relies on the fact that, by grayscale opening, one removes from an image the light structures that cannot hold the structuring element. Subtracting the opened image from the original one yields an image where the objects that have been removed by opening clearly stand out. This image can then easily be thresholded (see Fig. 1.11). Formally, the following definition can be proposed:

Definition 15 The top hat $TH_\gamma(I)$ of a grayscale image I with respect to opening γ is given by:

$$TH_\gamma(I) = I - \gamma(I), \quad (1.33)$$

where $-$ stands for the pixelwise subtraction operator.

Note that in the above definition, the opening γ does not have to be based on a structuring element, but could very well be an algebraic

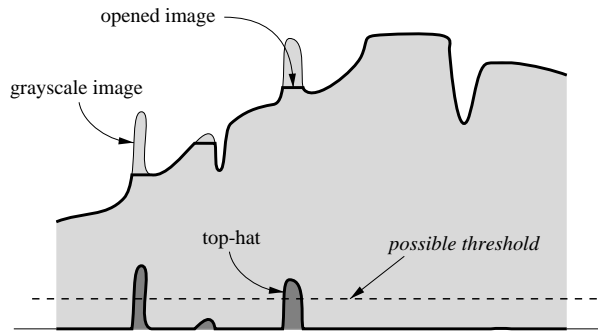


Figure 1.11 *Extraction of small bright structures (“peaks”) from an uneven background using top hat followed by thresholding.*

opening, as defined in Definition 12. The dual top hat TH^* can also be defined as the algebraic difference between a closing and the identity operator. It is used to extract small dark patterns from grayscale images.

An application is shown in Fig. 1.12: Fig. 1.12a is a scanning electron microscopy image where the balls in the lower right corner are to be extracted. These being compact and light compared to the background around them, they are removed by an opening of size 2 (see Fig. 1.12b). After subtraction of (b) from (a), i.e. top hat (see Fig. 1.12c), these small balls stand out and this image can be easily thresholded into Fig. 1.12d. The desired balls (right side) can now be extracted as the balls contained in the largest connected component of the dilation of Fig. 1.12d. The dilated image is shown in Fig. 1.12e, and the resulting segmentation is shown in Fig. 1.12f.

Let us conclude this first section by looking back at the approach that led to an operator such as the top hat: it is defined as the algebraic difference between the identity operator and a grayscale opening, itself obtained as the composition of a dilation and an erosion*. This is typical of the morphological approach to image analysis: given a few basic operations (erosions and dilations), one can compose them, iterate them, or combine them using Boolean and algebraic operators, thus creating increasingly complex and sophisticated transformations, that can be used towards increas-

* Dilations and erosions can themselves be decomposed into maxima and minima of translations.

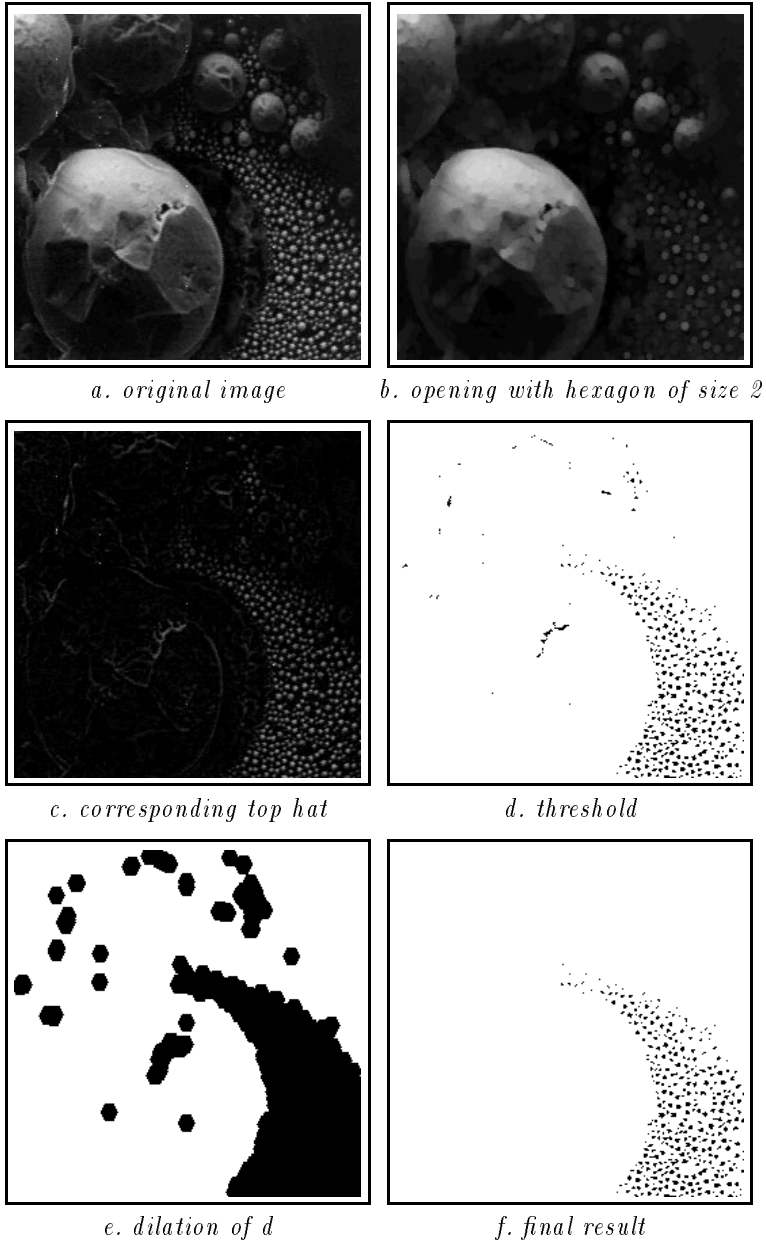


Figure 1.12 *Top-hat segmentation of SEM image.*

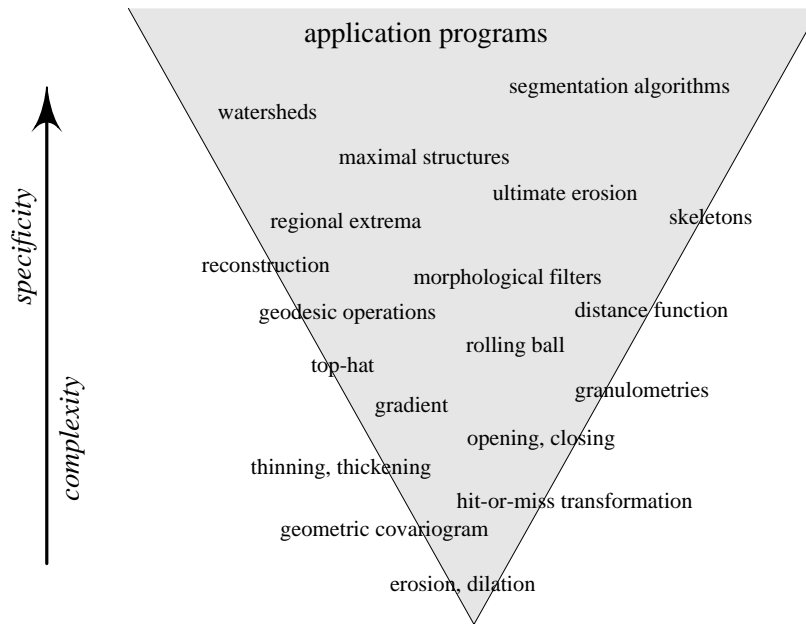


Figure 1.13 *Inverted pyramid of morphological operators.*

ingly specific tasks. This principle is illustrated by Fig. 1.13. A fair number of the operators listed in this illustration will be covered in the present chapter.

1.3 Advanced Morphological Segmentation Techniques

1.3.1 Introduction, Image Segmentation

The present section is primarily concerned with the *segmentation* of binary and grayscale images. In the image analysis world, the word *segmentation* refers to the decomposition or partitioning of an image into its meaningful “segments”, i.e., zones, or objects. For example, a typical segmentation task in remote sensing, consists of decomposing an aerial (satellite) image into urban areas and rural areas. Similarly, segmenting an image of biological material may come down to extracting individual cells and differentiating between cells and background.

Image segmentation therefore refers to a wide variety of image

analysis tasks. There is never a unique way to segment an image: the process is heavily goal-driven. Going back to the remote sensing example in the previous paragraph, for some applications, segmentation may come down to tagging each pixel depending on whether it is thought to belong to a rural region, or an urban region. For other applications however, one may need to detect and count all the different urban regions of the image: pixels tagged as “urban” will therefore need to be clustered together into connected regions, and these regions will need to be individually tagged. For yet other applications, it may be necessary to segment rural regions into, e.g., corn fields, wheat fields, potato fields, and the rest. It is therefore impossible to give a formal definition for “image segmentation”.

The region-contour duality should also be noted. Partitioning an image into different regions is strictly equivalent to extracting the contours of these regions or objects: these contours simply form the separating lines between regions. *Contour extraction*, which has traditionally been seen as a separate image processing task, is therefore nothing more than a different perspective on segmentation, and is covered in the present section.

While segmentation can take so many different aspects, this section mainly deals with object segmentation, i.e., with images of objects or particles (such as cells, beans, etc), where segmenting is equivalent to extracting the objects of interest. Specifically, the goal of particle segmentation is to partition the image in as many connected components as there are objects or regions to extract, plus one or more *background* regions. We distinguish between binary and grayscale particle segmentation: in the binary case, i.e. when the images under study are binary, the segmentation task consists of separating the touching or overlapping particles (e.g., see the coffee beans example of Fig. 1.3a). In the grayscale case, the segmentation task is equivalent to a contour extraction problem (e.g., in Fig. 1.3b, the contours of the electrophoresis spots have to be extracted as accurately as possible).

This section is not intended as a review of all the existing image segmentation methodologies. Instead, we focus on morphological segmentation tools and techniques: they are extremely powerful, easy to use, and suitable for a wide variety of applications. We begin with *geodesic* operators and illustrate their use for object extraction from grayscale images. The *grayscale reconstruction* operator described is the most versatile of these tools and is discussed at length. Grayscale *area openings and closings* are also described,

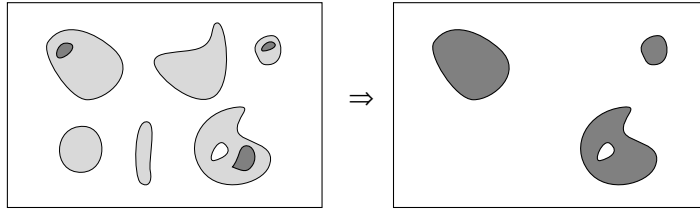


Figure 1.14 *Binary reconstruction from markers.*

and shown to provide a useful alternative to grayscale reconstruction in some segmentation applications. This progressively leads us to Section 1.3.4, where watershed segmentation techniques are described. We show that the watershed transformation is a compelling operation for the segmentation of both binary and grayscale images. Watersheds alone typically yield *oversegmented* results, but this issue is addressed using the grayscale reconstruction operator mentioned above, and this results in a consistent and widely applicable segmentation methodology. Finally, Section 1.3.5 describes the concept of *dynamics* and shows how, together with watersheds, it can be used for both automatic and interactive segmentation applications.

1.3.2 Grayscale Reconstruction and Applications to Segmentation

Geodesy, binary reconstruction

Let I and J be two binary images defined on the same discrete domain D and such that $J \subseteq I$. In terms of mappings, this means that: $\forall p \in D, J(p) = 1 \implies I(p) = 1$. J is called the *marker* image and I is the *mask*. Let I_1, I_2, \dots, I_n denote the n connected components of I .

Definition 16 *The reconstruction $\rho_I(J)$ of mask I from marker J is the union of the connected components of I which contain at least a pixel of J :*

$$\rho_I(J) = \bigcup_{J \cap I_k \neq \emptyset} I_k.$$

This definition is illustrated by Fig. 1.14. It is extremely simple, but gives rise to several interesting applications and extensions, as we shall see in the following.

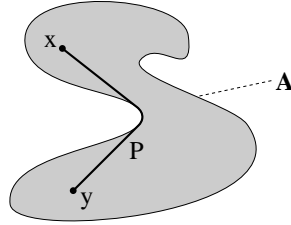


Figure 1.15 Geodesic distance $d_G(x, y)$ within a set A .

This binary reconstruction operator is known in the graphics community as “seedfill”. Another way to define it involves the notion of *geodesic distance*:

Definition 17 Given a set X (the mask), the geodesic distance between two pixels p and q in X is the length of the shortest paths joining p and q that are included in X .

Note that the geodesic distance between two pixels within a mask highly depends on the type of connectivity being used: geodesic distances in 4-connectivity are always larger than geodesic distances in 8-connectivity. Besides, an object that is connected in 8-connectivity may be made of several connected components in 4-connectivity. The notion of geodesic distance is illustrated by Fig. 1.15. Geodesic distance was introduced in the framework of image analysis by Lantuéjoul and Beucher (1981) and is at the basis of numerous morphological operators (e.g., see Lantuéjoul and Maisonneuve 1984). In particular, one can define geodesic dilations (and similarly, geodesic erosions) as follows:

Definition 18 Let $X \subset \mathbb{Z}^2$ be a discrete set and $Y \subseteq X$. The geodesic dilation of size $n \geq 0$ of Y within X is the set of the pixels of X whose geodesic distance to Y is smaller or equal to n :

$$\delta_X^{(n)}(Y) = \{p \in X \mid d_X(p, Y) \leq n\}.$$

Note that the notations used in the above definition conflict somewhat with the definitions used in Section 1.2: in the present context, subscript X in δ_X refers to the *mask* X within which the geodesic dilation is being performed. The grid G (4-, 6-, or 8-connected) is supposed to be fixed.

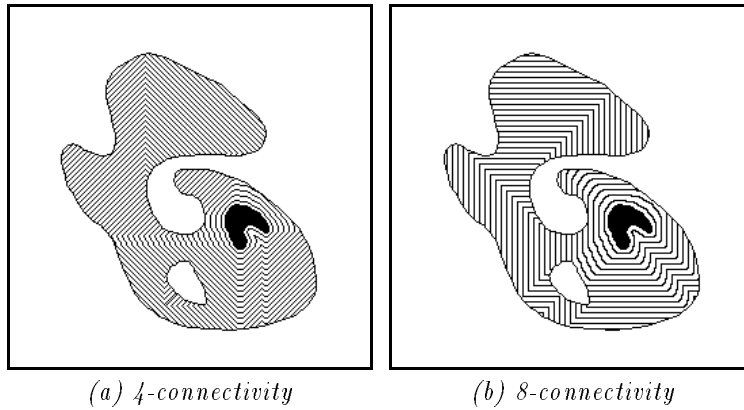


Figure 1.16 *Boundaries of the successive geodesic dilations of a set (black) within a mask.*

From this definition, it is obvious that geodesic dilations are extensive transformations, i.e. $Y \subseteq \delta_X^{(n)}(Y)$. In addition, geodesic dilation of a given size n can be obtained by iterating n elementary geodesic dilations:

$$\delta_X^{(n)}(Y) = \underbrace{\delta_X^{(1)} \circ \delta_X^{(1)} \circ \dots \circ \delta_X^{(1)}}_{n \text{ times}}(Y). \quad (1.34)$$

Fig. 1.16 illustrates successive geodesic dilations of a marker inside a mask, using 4- and 8-connectivity. The elementary geodesic dilation can itself be obtained via a standard dilation of size one followed by an intersection:

$$\delta_X^{(1)}(Y) = \delta_B(Y) \cap X. \quad (1.35)$$

This last statement only holds for *elementary* (i.e., unit-size) geodesic dilations.

When performing successive elementary geodesic dilations of a set Y inside a mask X , the connected components of X whose intersection with Y is non empty are progressively flooded. The following proposition can thus be stated:

Proposition 19 *The reconstruction of X from $Y \subseteq X$ is obtained by iterating elementary geodesic dilations of Y inside X until sta-*

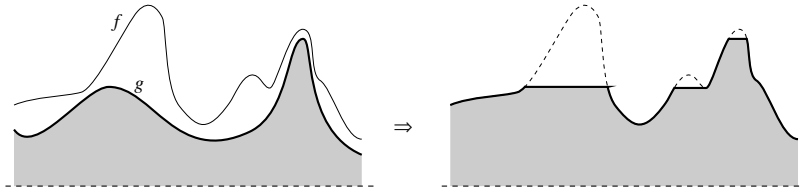


Figure 1.17 *Grayscale reconstruction of mask f from marker g .*

bility. In other words:

$$\rho_X(Y) = \bigcup_{n \geq 1} \delta_X^{(n)}(Y). \quad (1.36)$$

This proposition, which is easy to prove, forms the basis of one of the simplest algorithms for computing geodesic reconstructions. In addition, although equation 1.36 appears arcane at first, its extension to grayscale is straightforward.

Grayscale reconstruction

Binary geodesic reconstruction obviously is an increasing transformation in that it satisfies:

$$Y_1 \subseteq Y_2, X_1 \subseteq X_2, Y_1 \subseteq X_1, Y_2 \subseteq X_2 \Rightarrow \rho_{X_1}(Y_1) \subseteq \rho_{X_2}(Y_2). \quad (1.37)$$

Therefore, following the threshold superposition principle of Prop. 13, we can define *grayscale reconstruction* as follows (Vincent 1992a, Vincent 1993a).

Definition 20 (Grayscale reconstruction, first definition)

Let J and I be two grayscale images defined on the same domain, taking their values in the discrete set $\{0, 1, \dots, N\}$ and such that $J \leq I$ (i.e., for each pixel $p \in D_I, J(p) \leq I(p)$). The grayscale reconstruction $\rho_I(J)$ of I from J is given by:

$$\forall p \in D_I, \quad \rho_I(J)(p) = \max\{k \in [0, N] \mid p \in \rho_{T_k(I)}(T_k(J))\}. \quad (1.38)$$

Fig. 1.17 illustrates this transformation. Just like binary reconstruction extracts those connected components of the mask which are marked, grayscale reconstruction extracts the *peaks* of the mask which are marked by the marker-image.

Definition 20 does not provide any interesting computational method to determine grayscale reconstruction in digital images. Indeed, even if a fully optimized binary reconstruction algorithm is used, one has to apply it 256 times to determine grayscale reconstruction for images on 8 bits! Therefore, it is most useful to introduce this transformation using the concept of geodesic dilations presented earlier.

Following the threshold decomposition principle, one can define the unit geodesic dilation $\delta_I^{(1)}(J)$ of grayscale image $J \leq I$ “under” I :

$$\delta_I^{(1)}(J) = \delta_B(J) \wedge I, \quad (1.39)$$

In this equation, \wedge stands for the pointwise minimum operator, which is the direct extension to grayscale of the concept of Boolean intersection.

The grayscale geodesic dilation of size $n \geq 0$ is then given by:

$$\delta_I^{(n)}(J) = \underbrace{\delta_I^{(1)} \circ \delta_I^{(1)} \circ \dots \circ \delta_I^{(1)}}_{n \text{ times}}(J). \quad (1.40)$$

This leads to a second definition of grayscale reconstruction:

Definition 21 (Grayscale reconstruction, second definition)

The grayscale reconstruction $\rho_I(J)$ of I from J is obtained by iterating grayscale geodesic dilations of J “under” I until stability, i.e.:

$$\rho_I(J) = \bigvee_{n \geq 0} \delta_I^{(n)}(J).$$

It is straightforward to verify that both this definition and definition 20 correspond to the same transformation.

Similarly, the elementary geodesic erosion $\varepsilon_I^{(1)}(J)$ of grayscale image $J \geq I$ “above” I is given by

$$\varepsilon_I^{(1)}(J) = \varepsilon_B(J) \vee I, \quad (1.41)$$

where \vee stands for the pointwise maximum. The grayscale geodesic erosion of size $n \geq 0$ is then given by:

$$\varepsilon_I^{(n)}(J) = \underbrace{\varepsilon_I^{(1)} \circ \varepsilon_I^{(1)} \circ \dots \circ \varepsilon_I^{(1)}}_{n \text{ times}}(J). \quad (1.42)$$

We can now define the *dual* grayscale reconstruction in terms of geodesic erosions:

Definition 22 (Dual reconstruction) Let I and J be two grayscale images defined on the same domain D_I and such that $I \leq J$. The dual grayscale reconstruction $\rho_I^*(J)$ of mask I from marker J is obtained by iterating grayscale geodesic erosions of J “above” I until stability is reached:

$$\rho_I^*(J) = \bigwedge_{n \geq 1} \varepsilon_I^{(n)}(J).$$

Application to Segmentation

We will see in Section 1.3.4 that together with the watershed transformation, grayscale reconstruction forms one of the most powerful morphological tools for segmentation of binary and grayscale images. However, even without involving watersheds, grayscale reconstruction proves to be very useful for some segmentation applications.

For example, Fig. 1.18 represents an angiography of blood vessels in the eye, in which *microaneurisms* have to be detected. How should we approach this task? For every image analysis problem, a wise first step is to list all the pieces of information available on the collection of images to be processed. Indeed, as mentioned earlier, segmentation is a goal driven task, and completely general segmentation procedures simply do not exist. Here, our knowledge can be summarized as follows:

1. microaneurisms are small, compact, and brighter than the background,
2. blood vessels are thin, elongated, and brighter than the background,
3. microaneurisms are disconnected from the network of blood vessels,
4. microaneurisms are preferentially located in the dark areas of the image.

In general, such a list cannot be established from a single image, and a relatively large sample is required. However, subject again to the limitations of a single chapter, we content ourselves with the image shown in Fig. 1.18

Using element of knowledge number 1 above, one could think of extracting microaneurisms using a standard *top hat* transform (See Section 1.2.3). An opening of Fig. 1.18 by a small disk is shown in Fig. 1.19a, and the corresponding top hat is shown in

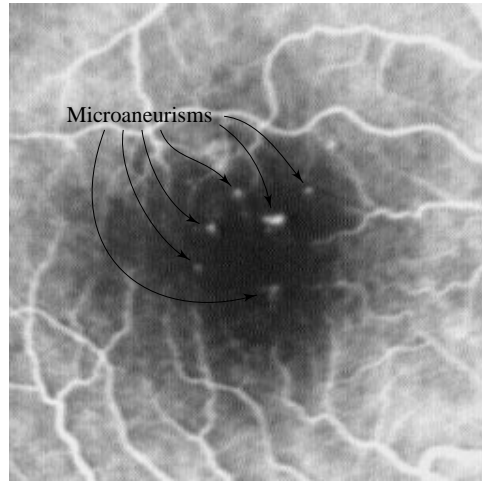
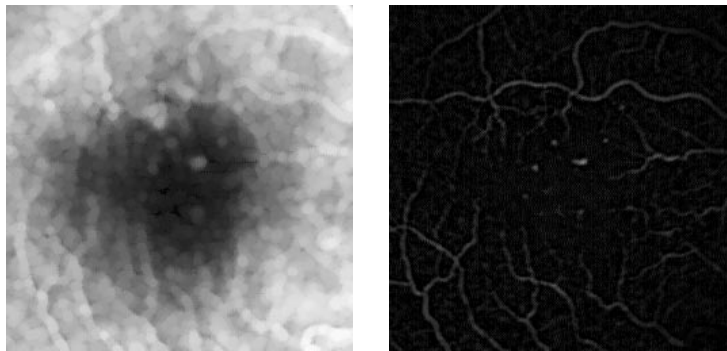


Figure 1.18 *Angiography showing eye blood vessels and microaneurisms.*



(a) *opening with small disk* (b) *corresponding top hat*

Figure 1.19 *Inadequacy of the top hat for aneurism extraction.*

Fig. 1.19b. This operation is clearly inappropriate here, because it extracts bright thin structures indiscriminately. The background of Fig. 1.19b does get normalized in the process, but the aneurisms have *not* been separated (segmented) from the blood vessels.

This preliminary experiment indicates that additional domain-specific knowledge should be used. The key pieces of information

here are number 2 and 3: the aneurisms are not touching the blood vessels, and these vessels are—as one could expect—elongated. To take this into account, we use a series of morphological openings of Fig. 1.18 using line segments in different orientations. These segments are chosen to be longer than any possible aneurism, so that all the aneurisms are removed by any such opening. On the other hand, the vessels being elongated, there is at least one orientation for which they are not completely removed by opening. After taking the supremum of these different openings, one obtains the result shown in Fig. 1.20a.

One can show that this maximum of openings using different structuring elements (specifically here: line segments in different orientations) is in fact an *algebraic opening* (see Section 1.2.2). It removes all the aneurisms, but the size of the line segments used is such that the network of blood vessels is not completely removed: plenty of *markers* remain, as can be seen in Fig. 1.20a. A grayscale reconstruction operation of Fig. 1.18 from Fig. 1.20a therefore enables us to reconnect (reconstruct) the blood vessels from their markers. But since the aneurisms are not connected to these vessels, they are not recovered by this operation. The result of this step is shown in Fig. 1.20b. Note that this operation of algebraic opening followed by a reconstruction is also an algebraic opening, sometimes referred to as *opening by reconstruction*. By subtracting this image from the original image, one obtains Fig. 1.20c, in which the aneurisms have been nicely extracted together with some extraneous small objects.

If we were to threshold the image of Fig. 1.20c, we would still obtain an imperfect result. At this point, we have to take into account piece of information number 4: aneurisms are primarily located in the darker areas of the image. We could extract these dark areas (here the central region of the image) by simply smoothing, then thresholding Fig. 1.19a; after thresholding Fig. 1.20c and intersecting the result with the extracted dark regions, we would be able to extract the desired aneurisms.

However, this technique, which involves *two* thresholding operation, would probably not be optimally robust. It may work adequately on this particular image, but is likely to fail when used on other images of the same type. As a general rule, one should:

- use as few fixed parameters as possible: fixed parameters limit

the class of images on which the algorithm is likely to perform adequately,

- postpone thresholding operations as much as possible: indeed, once a gray image is thresholded into a binary one, the loss of information is huge and irreversible.

Our solution here is to perform a large opening of the original image (opening with a large isotropic element) to extract a background image, and to invert this image. The resulting image, shown in Fig. 1.20d, can be regarded as a fuzzy set: the brighter the pixel, the more likely it is that this pixel belongs to the dark image regions of the original image. By doing a pixelwise multiplication of this image with Fig. 1.20c, we enhance the candidate aneurisms of Fig. 1.20c in proportion to their likelihood of being part of a dark image region. The result is shown in Fig. 1.20e. A simple threshold of this image finally provides the result shown in Fig. 1.20f.

1.3.3 Grayscale Area Openings and Closings

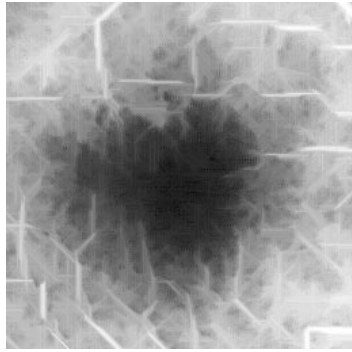
Background

In this section, we focus on the new concepts of area openings and closings, also called openings and closings *by area* (Vincent 1992b, Vincent 1993b). For a variety of segmentation applications, these operations provide a compelling alternative to openings followed by reconstruction.

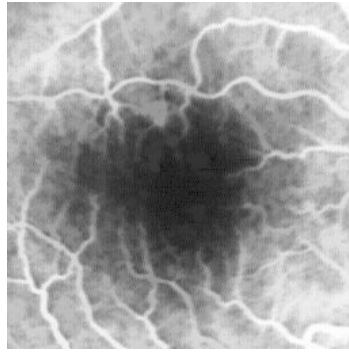
Let us begin with a classic image analysis preprocessing problem: filtering out small light (respectively dark) particles from grayscale images without damaging the remaining structures. Often, simple morphological openings (respectively closings) with disks or approximations of disks, are adequate for this task. However, when the structures that need to be preserved are elongated objects, they can be either completely or partly removed by such operations.

Consider for example Fig. 1.21a, which represents a microscopic image of a metallic alloy. It is “corrupted” by some black noise that one may wish to remove[†]. As shown in Fig. 1.21b, a closing of this image with respect to the elementary ball of the 8-connected metric, unit square S , severely damages most of the inter-grain lines, while still preserving some of the largest bits of noise (e.g., the blobs in the bottom right and left corners).

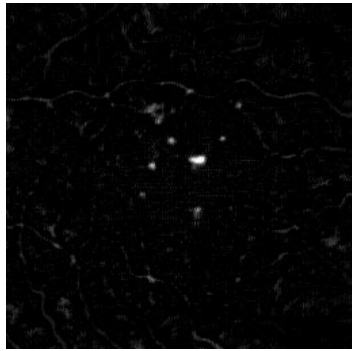
[†] Note that part of what is called noise here is the intra-grain texture.



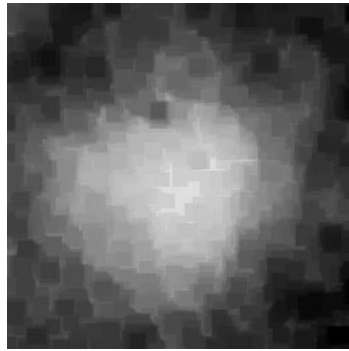
(a) maxima of linear opening



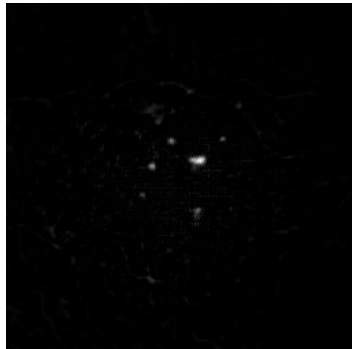
(b) reconstruction from (a)



(c) subtracting (a) from orig image



(d) inverted background



(e) pixelwise multiplication of (c) and (d) (f) threshold (dilated)



Figure 1.20 Extraction of micro-aneurisms using grayscale reconstruction

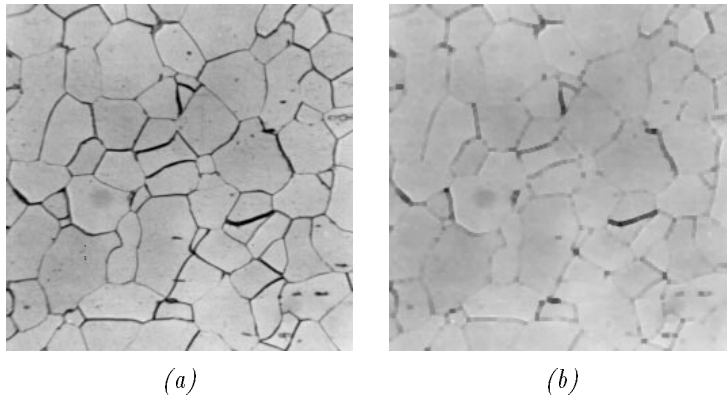


Figure 1.21 (a) *Microscopic image of a metallic alloy; (b) morphological closing of (a) by a unit-size square.*

In this context, algebraic openings (resp. closings) defined as maxima (resp. minima) of openings (resp. closings) with line segments in different orientations often preserve elongated structures better. Unfortunately, such operations are rather expensive computationally, and as illustrated by Fig. 1.22, they may remain unsatisfactory.

The remedy to this last problem is to increase the number of orientations of the used line segments, but this in turn increases the computational complexity of the algorithm. In addition, even with a large number of orientations, very thin lines might still end up broken. We already described in Section 1.3.2 how this problem can be addressed using the grayscale reconstruction operator. In the present section, we describe an alternative solution, using the concepts of *area openings* and *closings*. We show that these operators can outperform reconstruction-based algorithms and provide a new set of powerful segmentation tools.

Definition and Intuitive Interpretations

The easiest way *area openings* can be defined in the binary case is as follows:

Definition 23 (Binary area opening) *Let X be a set made of n connected components X_1, X_2, \dots, X_n . The area opening of pa-*

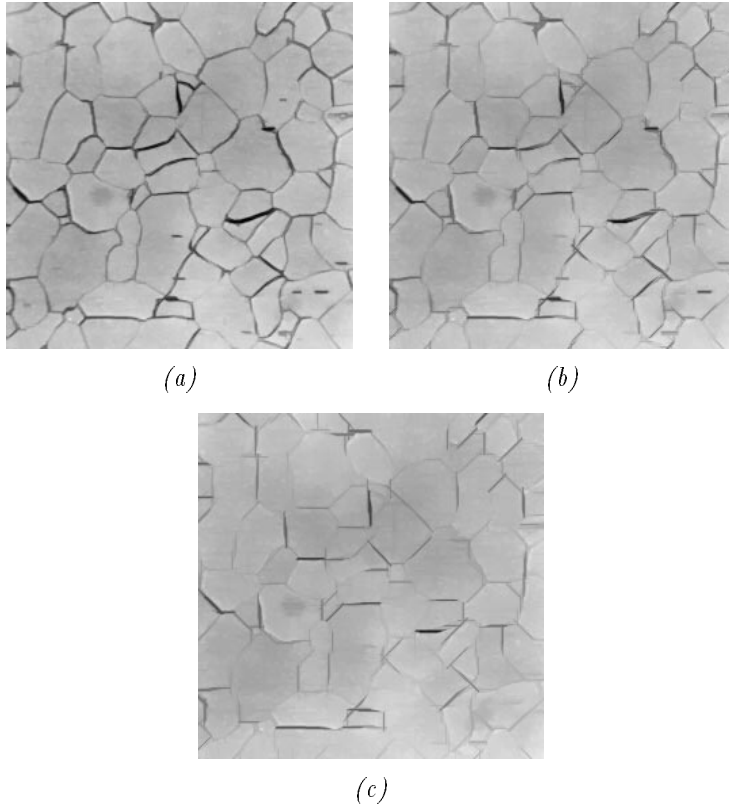


Figure 1.22 *Maxima of four linear openings of increasing size of Fig. 1.21a. For small sizes, noise remains after opening, and for larger sizes, the inter-grain boundaries start to get damaged.*

parameter $\lambda > 0$ of X is given by:

$$\gamma_\lambda^a(X) = \bigcup \{X_i \mid \text{Area}(X_i) \geq \lambda\}. \quad (1.43)$$

It follows that by area opening of size λ , we remove the connected components of X whose area is strictly smaller than λ , and preserve the others. Obviously, γ_λ^a is *increasing*, *idempotent*, and *anti-extensive*. It is therefore legitimate to call it an opening. By duality, the corresponding binary area closing ϕ_λ^a can easily be defined as the operator that fills the holes of X whose area is strictly smaller than λ .

These two operators, though very basic, are surprisingly useful for a number of binary image cleaning/filtering tasks. Moreover, following the threshold superposition principle of Prop. 13, they can be extended to grayscale, and the following definition can be proposed:

Definition 24 (Grayscale area opening) *The area opening $\gamma_\lambda^a(f)$ of a grayscale function f is given by:*

$$(\gamma_\lambda^a(f))(x) = \sup\{h \leq f(x) \mid \text{Area}(\gamma_x(T_h(f))) \geq \lambda\} \quad (1.44)$$

$$= \sup\{h \leq f(x) \mid x \in \gamma_\lambda^a(T_h(f))\}. \quad (1.45)$$

By duality, one similarly extends the concept of an area closing to grayscale. The interpretation of grayscale area openings and closings is relatively simple: a grayscale area opening basically removes from the image all the light structures which are “smaller” than the size parameter λ , whereas the area closing has the same effect on dark structures. Let us stress that the word *size* exclusively refers here to an area (or number of pixels in the discrete case).

In order to get a better intuitive understanding of grayscale area openings and closings, the following theorem can be stated:

Theorem 25 *Denoting by \mathcal{A}_λ the class of sets that are connected and whose area is greater than or equal to size parameter λ , the following equations hold:*

$$\gamma_\lambda^a = \bigvee_{B \in \mathcal{A}_\lambda} \gamma_B, \quad (1.46)$$

$$\phi_\lambda^a = \bigwedge_{B \in \mathcal{A}_\lambda} \phi_B. \quad (1.47)$$

In the discrete case, any connected set of area greater or equal to a positive integer λ contains a connected set of area equal to λ . We can therefore be more specific and state:

$$\gamma_\lambda^a(X) = \bigcup\{\gamma_B(X) \mid B \text{ connected, Area}(B) = \lambda\}, \quad (1.48)$$

$$\phi_\lambda^a(X) = \bigcap\{\phi_B(X) \mid B \text{ connected, Area}(B) = \lambda\}. \quad (1.49)$$

The above equations are valid in the grayscale case as well. This leads to a different understanding of area openings (resp. closings). As maxima of openings (resp. minima of closings) with all possible connected elements of a given area, they can be seen as *adaptive*: at every pixel location, the structuring element adapts its shape

to the local image structure in order to “remove as little as possible”. In particular, this means that bright elongated structures are usually very well preserved through area openings. Respectively, dark elongated structures, however fine, are typically unchanged by area closing.

Applications of Area Openings and Closings

In spite of their apparent complexity, area openings and closings lend themselves very well to efficient implementations, without which they would remain mostly useless (Vincent 1993b). They have two main types of application: filtering and segmentation.

In section 1.3.3, we described a filtering problem that was hard to address with traditional openings and combinations thereof. Given how thin the grain boundaries in Fig. 1.21a are, it is clear that a good filtering solution must use operators involving connectivity. Experiments were conducted using two different techniques:

1. minima of linear closings followed by dual reconstruction,
2. direct area closing.

As shown in Fig. 1.23, the results provided by these two techniques look very similar. However, the area closing adapts to local image structure better than the reconstruction-based method, and therefore better preserves grain boundaries. This is illustrated by Fig. 1.23c, which is the thresholded algebraic difference between Fig. 1.23a and Fig. 1.23b: the ON pixels in this image show where the area closing outperforms the reconstruction-based method. In addition, technique number 2 above is a straightforward one-step operation, and is about one order of magnitude faster than technique number 1.

It is perhaps for segmentation that area openings and closings are most interesting. We saw in Section 1.2.3 that standard grayscale morphological openings led to a simple and powerful segmentation tool, the *top hat* transformation. In Section 1.3.2, we extended this concept to openings based on maxima of linear openings followed by reconstruction. Now, this approach extends to area openings and closings, and proves to be powerful and straightforward.

Let us revisit the problem of microaneurisms extraction from the angiography of Fig. 1.18. Instead of using linear openings followed by reconstruction, we can directly perform an area opening on this image. If we choose the size parameter in this operation to be larger than the area of any possible aneurism, we are guaranteed

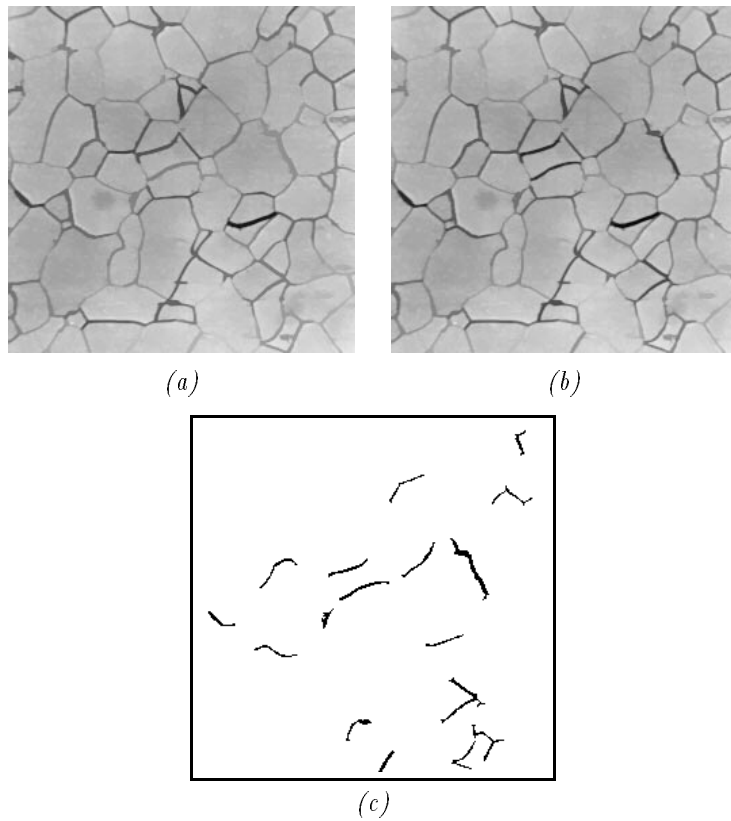
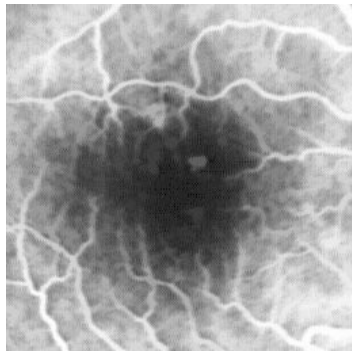


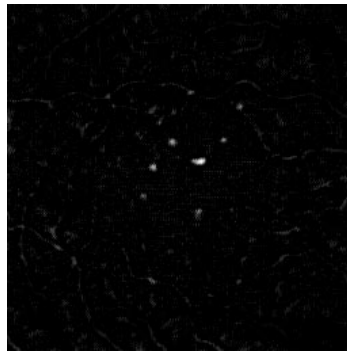
Figure 1.23 (a) Result of reconstruction-based technique for filtering Fig. 1.21a from. (b) Area closing of Fig. 1.21a. (c) Zones where technique 2 outperforms technique 1, and preserves grain boundaries substantially better.

that all the aneurisms will be removed by area opening. However, except for occasional bright spikes inside them, the blood vessels will be perfectly preserved: indeed, they form a connected network whose overall area is much larger than the area of any aneurism. The result of this area opening operation is shown in Fig. 1.24a. A pixelwise subtraction of this image from the original image of Fig. 1.18 leads to Fig. 1.24b. This operation can be called *area top hat* or *top hat by area*.

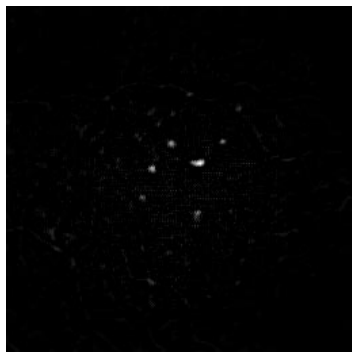
The aneurisms are now clearly visible, but some other small



(a) area opening of size 60 pixels



(b) area top hat



(c) enhancement of (b)



(d) extracted microaneurisms

Figure 1.24 Use of area opening in the detection of microaneurisms from angiographic imagery.

structures not located on the dark image areas are also present. To get rid of them and extract the final aneurisms, we use the same technique used earlier: combine Fig. 1.24b with an image of the background in order to enhance the structures in proportion to the darkness of the background around them. This leads to Fig. 1.24c, which can now be thresholded into Fig. 1.24d, the final result of this segmentation.

1.3.4 Watershed segmentation

The segmentation problems addressed thus far in this section are in fact “degenerate cases”: what really matters with the angiography of Fig. 1.18 is that every aneurism in it gets detected. However, extracting the exact outline of each detected aneurism is not needed here. This is far from being always the case, and in the present section, we consider problems where extracting object or region boundaries as accurately as possible is critical.

Let us start with binary segmentation, and consider the image shown in Fig. 1.3a. This image shows a number of coffee beans, many of which are touching or slightly overlapping. In order to measure for example the area, length, and width of each bean, it will be necessary to extract the separating lines between the beans, i.e., to segment this image. Note however that full segmentation is not always required:

- If counting the beans is all that is needed, then extracting the outline of each bean is not necessary. Individually detecting each bean (e.g., its centroid) would be sufficient, and is a simpler problem. Another approach would be to use statistical models such as Boolean models (Serra 1988), from which a bean count estimate could be derived.
- If a rough estimate of dominant bean size is all that is needed, then segmentation may not be necessary either, and one could simply use *granulometries*, as described in Section 1.4.

In the present case, the segmentation problem comes down to separating touching beans. To address this, recall that the erosion operation not only makes object smaller, but tends to disconnect touching object. Therefore, by performing erosions of increasing size on the coffee bean image of Fig. 1.3a, we progressively disconnect the coffee beans as the “bridges” between them are eroded away (See Fig. 1.25). Unfortunately, since the degree of overlap of these beans is not uniform throughout the image, they get disconnected for different erosion sizes. In other words, a single erosion size is not sufficient to disconnect the beans from each other, while not completely eroding away any bean.

To deal with this problem, a kind of “multiscale erosion” operator is used, called the *distance function*. Recall that B denotes the unit-size ball for the grid being used (diamond D in 4-connectivity, hexagon H in 6-connectivity, square S in 8-connectivity). The distance is defined as follows:

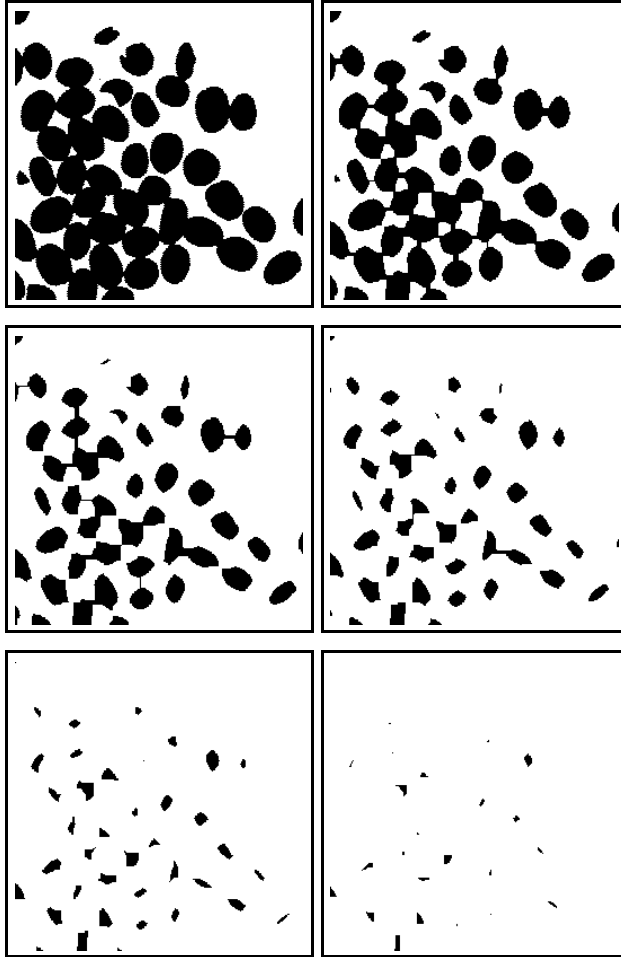


Figure 1.25 *Successive erosions of the coffee beans image*

Definition 26 *The distance function (or distance transform) $dist_X$ associated with a set X maps each pixel p of X to the smallest n such that p does not belong to the erosion of size n of X :*

$$dist_X(p) = \min\{n > 0 \mid p \notin \varepsilon_{nB}(X)\}. \quad (1.50)$$

The distance transform provides a synthetic representation of the information contained in all the successive erosions of X : by

thresholding it at level n , one simply obtains the erosion of size $n - 1$ of X :

$$\forall n > 0, \quad T_n(\text{dist}_X) = \varepsilon_{(n-1)B}(X). \quad (1.51)$$

One can also easily prove that the value of each pixel p in the distance function is equal to the distance between this pixel and the background X^C for the grid considered, hence the name *distance function*.

The distance function is an increasing operator that is unusual in that it maps binary images (sets) onto grayscale images. It has been extensively studied in literature, and thousands of papers have been published on different flavors of distance functions, distance function algorithms, applications of distance functions, etc. Some of the reference papers on distance functions include Rosenfeld and Pfaltz (1966, 1968), Borgefors (1984, 1986).

Applying this transformation to the coffee beans image results in the distance function shown in Fig. 1.26. Taking a topographic analogy—which we will do often from now on—each bean has been turned into a volcano-shaped mountain (or, if the reader prefers, a chinese hat); the valley-lines between these mountains correspond to the lines along which beans got separated through the original erosion process. These valleys form the lines that will provide the desired segmentation. The issue is: how should these lines be detected? The *watershed transformation* turns out to provide an outstanding way to answer this question.

The concept of watersheds is based upon such notions as *regional extrema* and *catchment basins*, which we briefly recall now.

Definition 27 (Regional maximum) *A regional maximum M of a grayscale image I is a connected components of pixels with a given value h (plateau at altitude h), such that any pixel in the neighborhood of M has a strictly lower value.*

Regional maxima constitute an important morphological topic and are extensively discussed in (Maisonneuve 1992). They should not be mistaken for *local* maxima: a pixel p of I is a local maximum for grid G if and only if its value $I(p)$ is greater than or equal to that of any of its neighbors. All the pixels belonging to a regional maximum are therefore themselves local maxima, but the converse is not true: for example, a pixel p belonging to the inside of a plateau is a local maximum, but the plateau may have neighboring pixels of higher altitude and thus not be a regional maximum.

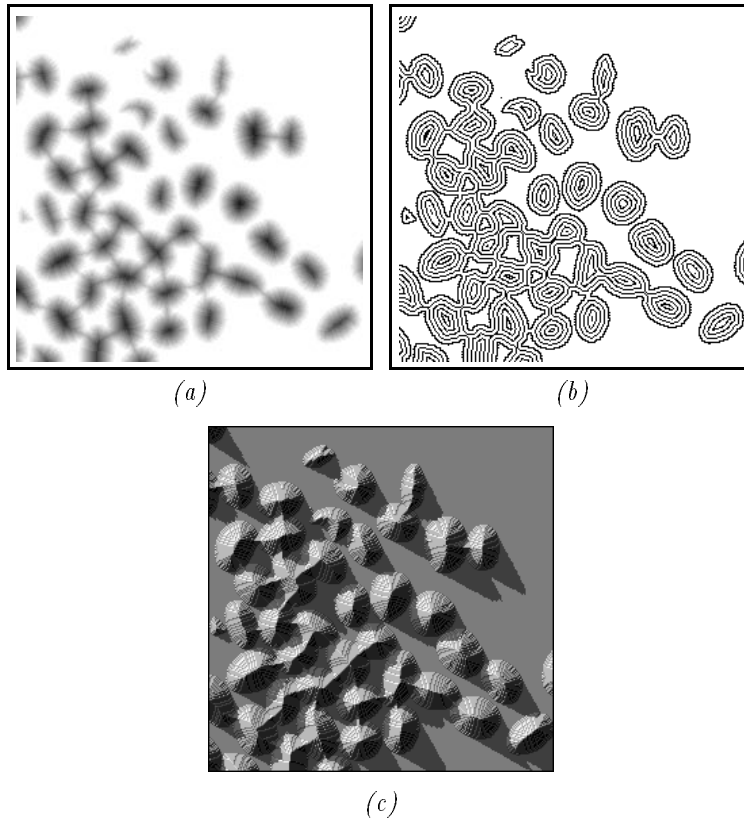


Figure 1.26 (a) Distance function of the coffee beans image. Dark pixels correspond to large values of the distance function, which was here computed using the Euclidean metric for greater accuracy (Vincent 1991a). (b) Level lines of this function. (c) Representation of this distance function as a relief, using artificial shading and shadowing.

Determining the regional maxima (resp. minima) of a grayscale image is relatively easy and several algorithms have been proposed in literature, some of which are reviewed in (Vincent 1990, 1993a). One of the most efficient methods makes use of grayscale reconstruction and is based on the following proposition:

Proposition 28 *The (binary) image $M(I)$ of the regional maxima of I is given by:*

$$M(I) = I - \rho_I(I - 1).$$

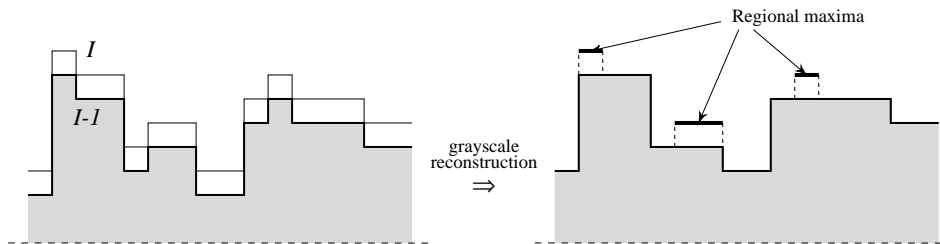


Figure 1.27 *Extracting the regional maxima of a grayscale image I by reconstruction of I from $I - 1$.*

A proof for this result can be found in . It is illustrated by Fig. 1.27. By duality, a similar technique can be derived, enabling to extract regional minima through dual grayscale reconstruction. In Section 1.3.5, we will see that this proposition forms the basis of the concept of *dynamics*.

Now, let us again regard grayscale image I as a topographic surface. If a drop of water falls at a pixel p of this relief, it will slide along the relief, following some steepest slope path, until it finally reaches one of the (regional) minima. We define the *catchment basin* $C(m)$ associated with a minimum m of I as follows:

Definition 29 (Catchment basin) *The catchment basin $C(m)$ associated with a (regional) minimum m of a grayscale image I , regarded as a topographic relief, is the set of the points p such that a drop falling on p slides along the surface until it reaches m .*

This definition, while not very formal, has the advantage of being intuitive. Obviously, catchment basins provide a tessellation of the space, and by definition, there is a one-to-one correspondence between minima and catchment basins. The crest-lines separating different basins are called *watersheds lines* or simply *watersheds*.

Definition 30 (Watersheds)

The watersheds (lines) of a grayscale image I are the lines that separate the different catchment basins of I .

These notions are illustrated by Fig. 1.28. Going back to our binary segmentation problem, the distance function $dist_X$ exhibits regional maxima roughly located near the centroid of each bean. These maxima of the distance function provide *markers* of each bean, and form what is known as the *ultimate erosion* of the original binary image. They are separated by the valley lines that we are

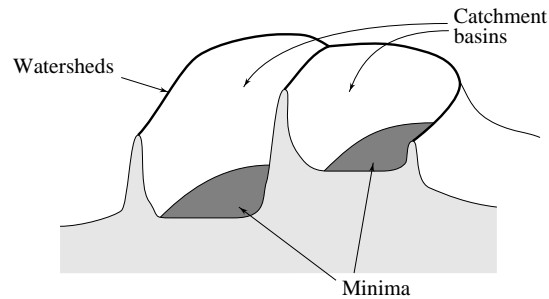


Figure 1.28 *Regional minima, catchment basins, and watershed lines.*

trying to detect. If we invert this relief, by duality, regional maxima are turned into regional minima and valley lines are turned into crest lines. By extracting the catchment basins of $-dist_X$, we associate with each minimum (i.e., bean marker) its catchment basin. The catchment basins are separated by watershed lines located on the valley lines of the original distance function $dist_X$. In summary, binary segmentation is achieved using the watersheds of the inverted distance function, as illustrated by Fig. 1.29.

In practice, things are almost never that easy, and even our academic coffee bean segmentation problem cannot be segmented that straightforwardly. As illustrated by Fig. 1.30a, there is not one-to-one correspondence between coffee beans and regional maxima of the distance function: the centroid of each bean is marked by at least one such maximum, but some beans are multiply marked, and some maxima are even located in-between beans (an example of this can be seen in the lower left corner of the image). As a result, watershedding the inverted distance function produces an oversegmented result, as shown in Fig. 1.30b.

This problem is not insurmountable: the maximal regions of the distance function are rather obvious structures, and the reason why they sometimes produce more than one regional maximum each is due to the fact that the workspace is discrete. The distance value that gets assigned to each pixel may sometimes get rounded up or down, which typically results in unit gray level oscillations in the vicinity of the top of the maximal structures. Specifically, if a structure (i.e., a bean) produces multiple maxima, these maxima are all at the same altitude h ; in addition, for each pair of such maxima, there exists a path connecting them in such a way that

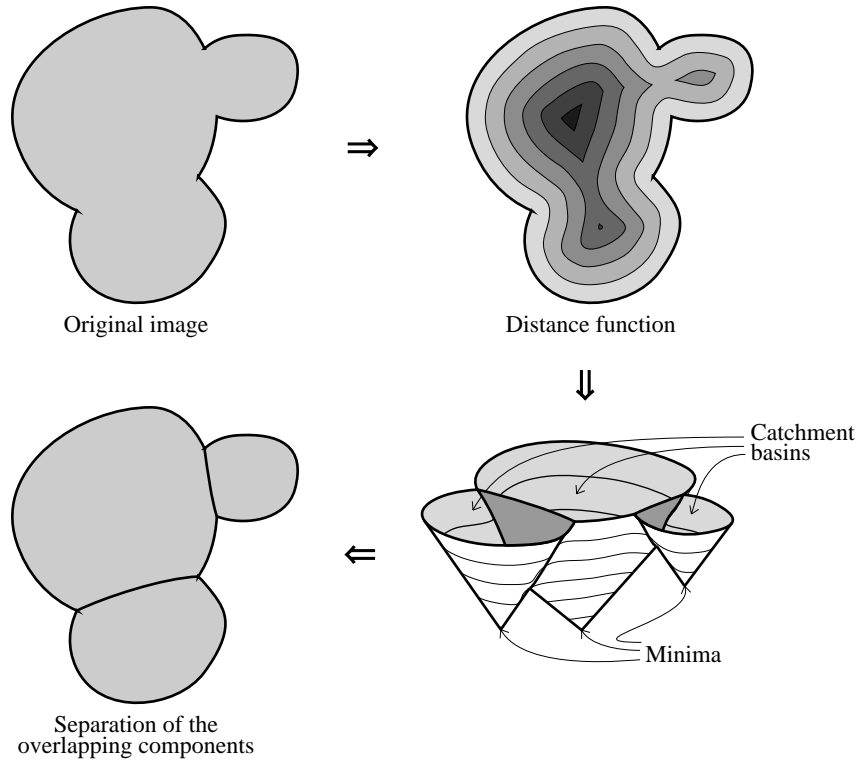
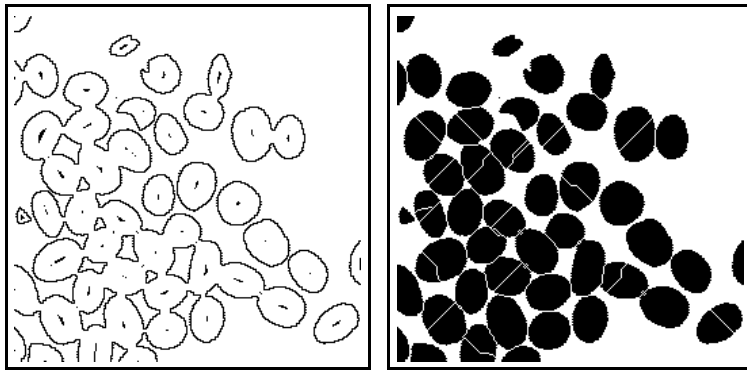


Figure 1.29 *Binary segmentation using catchment basins of the opposite of the distance function.*

the distance values along the path are all equal to h or $h - 1$. In other words, these maxima may be relatively far from each other in terms of distance (along the grid), but are very close in terms of gray-level.

In order to connect these maxima, the versatile grayscale reconstruction operator is used: the principle is to reconstruct $dist_X$ from $dist_X - 1$. In doing so, maxima marking the same structure are connected into one single blob. In addition, irrelevant maxima, such as the one in the bottom left corner of Fig. 1.30a, are removed in the process. This technique is illustrated in Fig. 1.31.

Going back to the coffee beans example, we can apply this method directly. Fig. 1.32a shows the modified distance function



(a) maxima of dist. fct. (b) watersheds of inverted dist. fct.

Figure 1.30 Direct watersheds of the inverted distance function produce an oversegmented result.

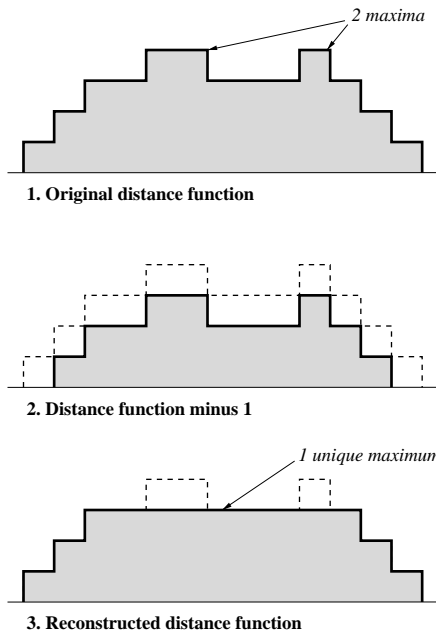
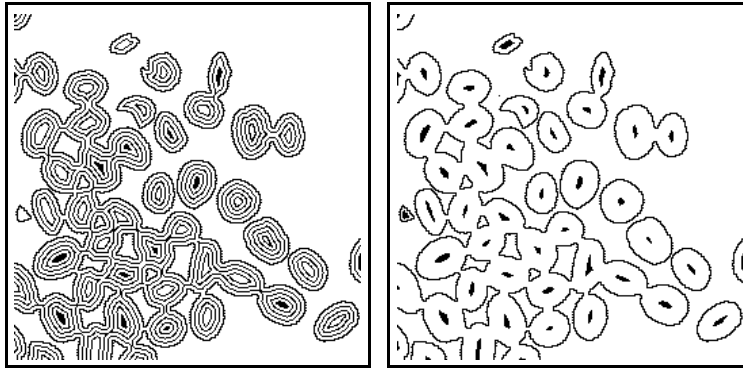
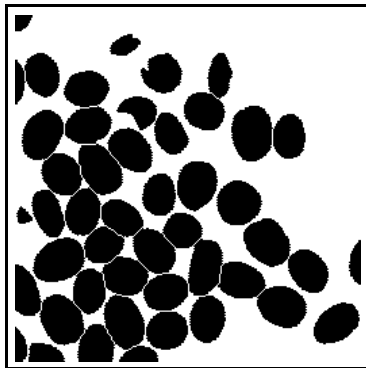


Figure 1.31 Connection of regional maxima using grayscale reconstruction of the distance function $dist_X$ from $dist_X - 1$.



(a) level lines of modified dist. fct. (b) maxima of (a)



(c) watersheds of inverted modified distance: final result

Figure 1.32 Watershed segmentation of coffee beans image.

obtained after grayscale reconstruction. Extracting the maxima of this function results in Fig. 1.32b: each bean is now marked once, and there is no extraneous marker left. Therefore, the watersheds of the inverted modified distance function provide the desired segmentation, as shown in Fig. 1.32c.

Watersheds stand out as a powerful morphological crest-line extractor. It is therefore most interesting to apply the watershed transformation to gradient images: indeed, the contours of a grayscale image can be viewed as the regions where the gray levels exhibit the fastest variations, i.e., the regions of maximal gradient. These regions form the *crest-lines of the gradient*. This

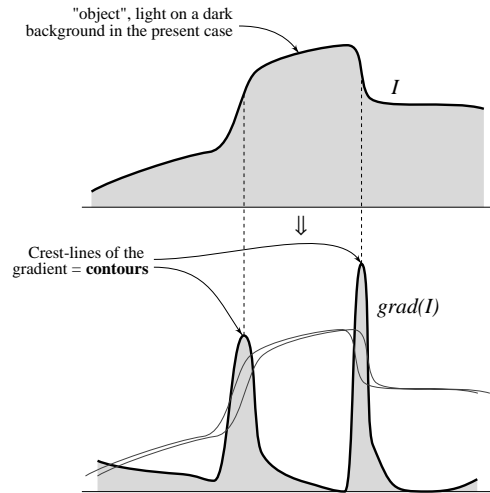


Figure 1.33 *Principle of grayscale segmentation via watersheds of the gradient.*

principle is illustrated by Fig. 1.33 and is at the basis of the use of watersheds for grayscale segmentation, as described and illustrated in (Beucher and Lantuéjoul 1979, Beucher 1982, Vincent and Beucher 1989, Vincent and Soille 1991).

Note that in morphology, the word *gradient* refers to an operation associating with each image pixel the *modulus* of its gradient—in the classical sense of the word. Most of the time, the gradient known as *Beucher's gradient* (Serra 1982) is used. It is obtained as the algebraic difference of a unit-size dilation and a unit-size erosion of image I :

$$grad(I) = (I \oplus B) - (I \ominus B).$$

However, depending on the type of image contours to be extracted, other gradients may be of interest: directional gradients, asymmetric gradients, regularized gradients, etc (Rivest, Soille and Beucher 1992).

The watershed transformation always provides *closed* contours and constitutes a very general approach to contour detection. However, rarely can it be used directly on gradient images without resulting in dramatic *over-segmentations*: the image gets partitioned in far too many regions, i.e., the correct contours are lost in a

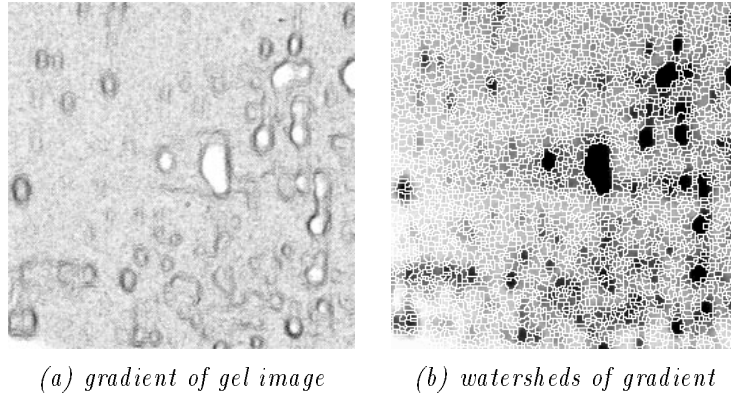


Figure 1.34 *Using the watershed operator directly on the original image's gradient typically results in extreme oversegmentation. In (b), the watershed lines are superimposed on the original gel image.*

large number of irrelevant ones. This problem is mainly due to noise in the data: noise in the original image results in noise in its morphological gradient, this in turn causing it to exhibit far too many regional minima. This directly translates into far too many catchment basins, i.e., over-segmentation.

In the rest of this section, we use the classic image of Fig. 1.3b to illustrate the methodology exposed[‡]. This image represents a two-dimensional electrophoretic gel, obtained through the migration of proteins in an electric field. Such gels are becoming increasingly useful in criminal cases, as they provide a means to attach a “genetic signature” with any individual. Here, our purpose is not only to detect all the dark spots in this image, but also to extract their outline as accurately as possible.

The standard morphological gradient of this image is shown in Fig. 1.34a. As mentioned above, if we simply compute the watersheds of Fig. 1.34a, the result is clearly disappointing, as proved by Fig. 1.34b. Indeed, the gradient exhibits a large number of minima, mainly due to the presence of noise in the original data. Note however that *all* the correct spot contours are present: they are just lost among hundreds of “illegal” contour elements.

[‡] This image is classic because it has been used in at least a dozen papers dealing with watershed segmentation.

Several approaches have been proposed in literature to overcome this over-segmentation: for example, some techniques remove arcs of the watersheds based on an integration of the gradient's gray values along them. Others, known as *region-growing* techniques, take the dual point of view and merge adjacent regions (i.e., catchment basins here) when the gray level of the original image over them is comparable. None of these techniques is satisfactory in that it is very difficult to incorporate to them knowledge specific to the collection of images under study.

An approach developed in the mathematical morphology community has proved to be a lot more robust and widely applicable to a range of image segmentation problems. This approach is as follows:

- Make use of image-specific knowledge (e.g., size, shape, location or brightness of the objects to extract) to design robust object marking procedures (Vincent and Beucher 1989, Vincent and Soille 1991, Beucher and Meyer 1992). This step of the segmentation can be completely different from one problem to another. Not only must each object be uniquely marked, the *background*—if any—is also a region, and needs to be marked as well. The idea behind this marker extraction step is that markers are generally easier to extract than actual object contours. By extracting markers first, we reduce the complexity of the problem and make it easier to incorporate domain-specific information into the segmentation.
- Use these markers to *modify the gradient image* on which watersheds are computed. Let us stress that the markers are not used to help in the postprocessing of the oversegmented watershed image, but instead, to alter the (gradient) image on which the watershed transformation is computed.

Let us describe this second step in more detail: let I denote the original grayscale image, let $J = \text{grad}(I)$ be its morphological gradient (or rather, a gradient chosen to best enhance the desired edges), and let M denote the binary image of markers previously extracted. The “modification” of J should result in a grayscale image J' with the following characteristics:

- its only regional minima are exactly located on the connected components of M (M is the set of “imposed” minima);
- its only crest-lines are the highest crest-lines of J that are located between the imposed minima.

The watersheds of J' would then be the highest crest-lines of $\text{grad}(I)$ that separate our markers, that is, the optimal contours with respect to our markers M and our gradient J .

The actual computation from J and M of an image J' with these characteristics has been classically achieved using a three-step process (Vincent and Beucher 1989):

1. Set to h_{\min} any pixel of J that is located on a marker, h_{\min} being chosen such that $\forall p, h_{\min} < J(p)$. This results in a new image J^* :

$$\forall p, \quad J^*(p) = \begin{cases} h_{\min} & \text{if } M(p) = 1 \\ J(p) & \text{otherwise.} \end{cases}$$

2. Create the following grayscale image M^*

$$\forall p, \quad M^*(p) = \begin{cases} h_{\min} & \text{if } M(p) = 1 \\ h_{\max} & \text{otherwise,} \end{cases}$$

where h_{\max} is chosen such that $\forall p, J(p) < h_{\max}$.

3. Use M^* to remove all the unwanted minima of J^* while preserving its highest crest-lines between markers. This is done using the dual grayscale reconstruction operation ρ^* :

$$\forall p, \quad J'(p) = \rho_{J^*}^*(M^*). \quad (1.52)$$

This process is illustrated by Fig. 1.35. The watersheds of the resulting image J' then provide the desired segmentation.

The whole procedure presented above is often referred to as *marker-driven watershed segmentation*. It is extremely powerful in a number of complex segmentation cases, where it essentially reduces the segmentation task to (1) the choice of a gradient and (2) the extraction of object markers (this latter task being itself very complex in some cases).

In practice, the two steps of gradient modification followed by watershed extraction can be folded into one. For example, the watershed algorithm introduced by Vincent and Soille (1991) can easily be modified to take a gradient image *and* a marker image as input, and directly produce the set of watershed lines of the modified gradient.

On our electrophoresis gel example, avoiding over-segmentation requires the prior extraction of correct spot markers. Since the spots constitute the dark part of the image, they can be interpreted as the image minima. However, once again, the noise in this

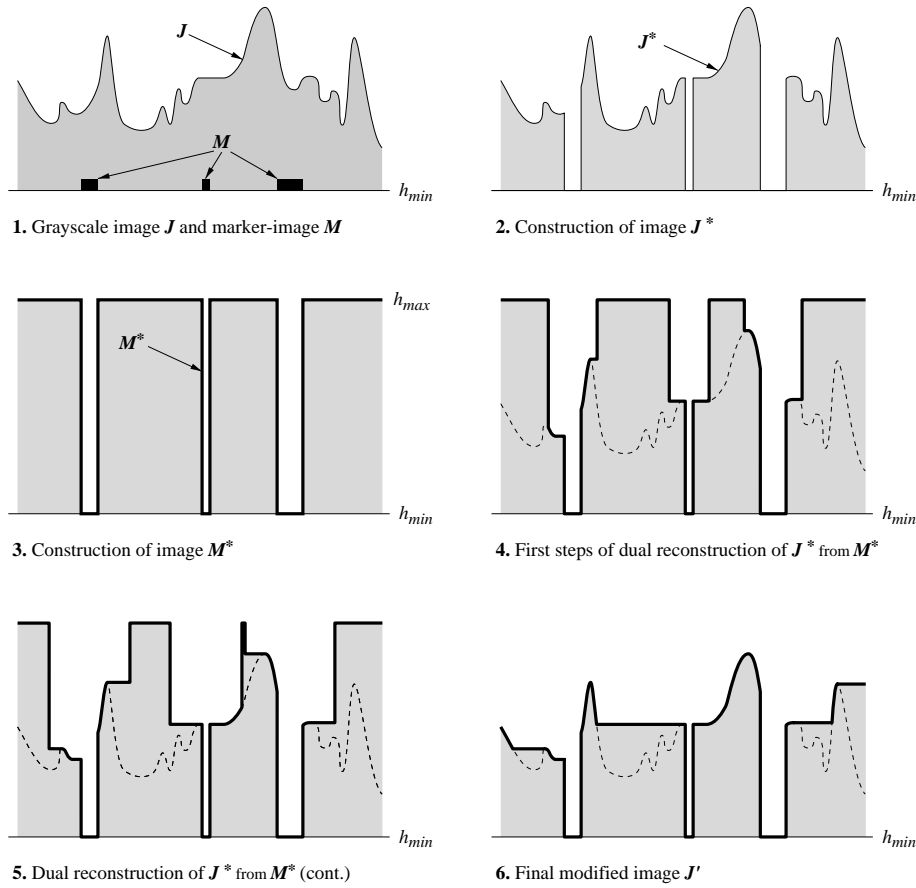


Figure 1.35 Use of dual grayscale reconstruction to “impose” a set M of minima to a grayscale image J .

data would make a direct minima extraction completely useless. Fortunately, filtering the original gel image using a standard morphological filter called an *alternating sequential filter (ASF)* (Serra 1988, Serra and Vincent 1992) is sufficient to produce an image whose minima correctly mark the spots, as shown in Fig. 1.36a.

In fact, this marker extraction step is followed by binary watershed segmentation in order to cut markers like the upper-right corner one, which clearly should mark two different spots. Arguably,

even after this, our set of markers is still not perfect, but it is good enough for our present purpose. Note that the quality of the final result is directly related to the quality of the initial marker extraction: poor markers automatically result in poor segmentation. (On the other hand, good markers do not absolutely guarantee that the final segmentation will be perfect.)

The background marker of the gel image is easily extracted as the set of the highest crest-lines of the original image that separate the spot markers. This is the best way to ensure that this marker is located on the brightest areas of the image and separates all the spot markers. Its determination is done in a similar way as gradient modification (see Eq. 1.52 and Fig. 1.35). It is shown in Fig. 1.36b.

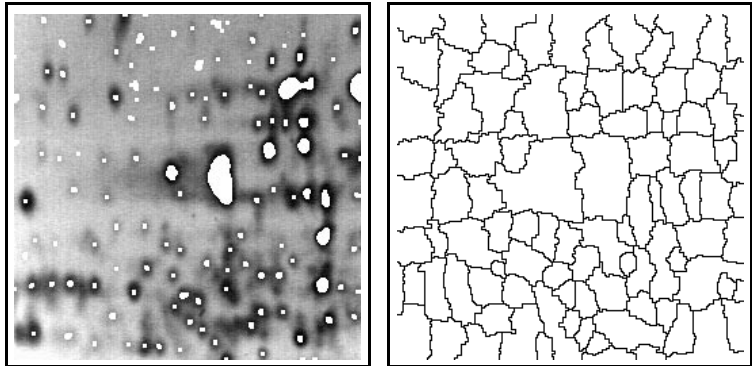
Both sets of markers are then combined in a final marker image, shown in Fig. 1.36c. It is used to modify the gradient of Fig. 1.34a, this resulting in Fig. 1.36d. The watersheds of the latter image provide the desired segmentation, as shown in Figs. 1.36e-f.

The result is in accordance with our expectations: each spot has a unique contour which is located on the inflexion points of the initial luminance function—i.e. the original image. Given the extracted set of markers, we found the optimal gradient crest-lines, i.e. the best possible segmentation for these markers and, to a lesser extent, this gradient. Notice that in a few places, the contours shown in Figs. 1.36e-f leave to be desired: this means that the gradient should have been more carefully designed for this particular segmentation problem (the standard morphological gradient was used here). Similarly, some spots have not been detected, and some appear to have been merged: this is directly related to the less-than-perfect marking method that was crafted here.

Numerous applications illustrating the power of this marker-driven watershed segmentation paradigm could be shown, but then again, a single chapter would not be sufficient. We refer the reader to the literature, and in particular to the following articles and chapters: Talbot and Vincent 1992, Vincent and Beucher 1989, Vincent and Masters 1992, Vincent 1993a, Vincent and Dougherty 1994.

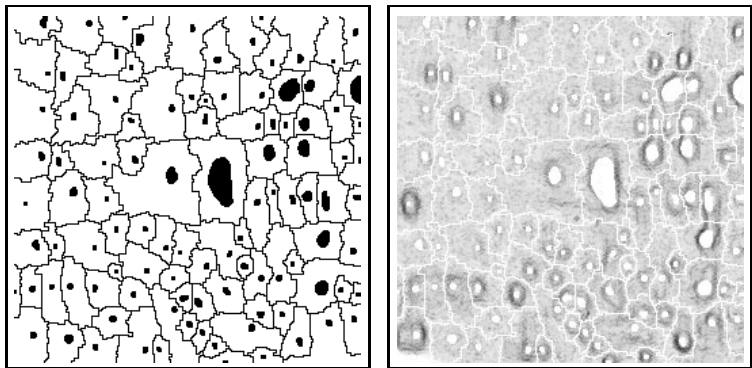
1.3.5 Dynamics and Hierarchical Watershed Segmentation

This section on state-of-the-art morphological segmentation techniques would not be complete without a brief summary on the concept of *dynamics*, its use for segmentation, and some of the



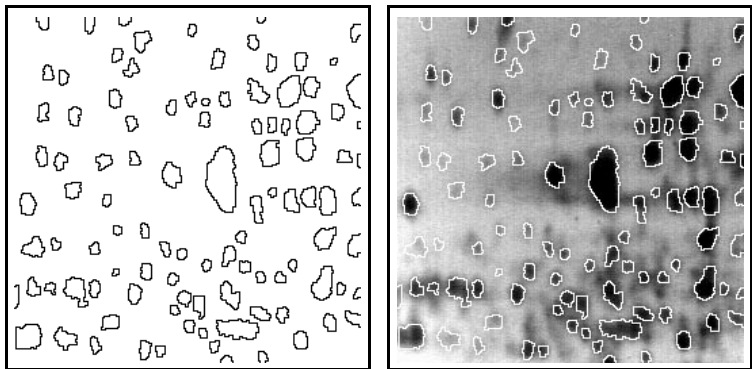
(a) minima of filtered image

(b) background marker



(c) complete set of markers

(d) modified gradient



(e) watersheds of modified gradient

(f) overlaid result

Figure 1.36 Watershed segmentation of electrophoresis gel image

derived transformations. Earlier in this section, the usefulness of regional *minima* and *maxima* was illustrated. Not only do maxima and minima provide means to extract object markers for binary and grayscale segmentation, they are also the basis for such concepts as catchment basins and watersheds.

In Section 1.3.4, we also showed how irrelevant maxima could be eliminated using grayscale reconstruction (see Prop. 28): the technique proposed there, based on reconstructing the image I from $I-1$, enabled us to eliminate maxima based on their relative height (brightness) with respect to other maxima. Specifically, maxima that are such that one can “walk” from one to the other on a path that never dips down more than one gray level got merged (see Fig. 1.27). In addition, if these maxima were such that one could reach a higher maximum by traveling along a path that dips down by at most one gray level, they were eliminated altogether.

Now, instead of reconstructing I from $I-1$, one could instead reconstruct from $I-2$. This would eliminate even more maxima. Ultimately, by reconstructing I from all the $I-n$, with $n = 0, 1, 2 \dots$, one can tag each maximum of an image by the first value n for which this maximum disappears. This is called the *dynamics* of this maximum. Alternatively, we can also state the following definition, originally proposed by Grimaud (1992):

Definition 31 *The dynamics of a regional maximum m of a grayscale image I is defined as:*

$$\text{Dyn}(m) = \min_P \{I(m) - I_{\min}(P)\} \quad (1.53)$$

where P is a path (connected set of pixel) joining m and a maximum higher than m , and where $I_{\min}(P)$ is the minimal value of image I along P .

In simpler terms, and using again a topographic analogy, the dynamics of m is equal to the minimal altitude loss that has to occur when traveling between m and a maximum of higher altitude. This is illustrated by Fig. 1.37. The dynamics of image *minima* is defined in a similar fashion.

This definition raises a couple of questions: first, what is the dynamics of the highest maximum in the image? According to the above definition, it is undefined. For consistency, we just need to make sure that it is higher than the dynamics of any other maximum in the image. Second: what happens with maxima of the same altitude? Indeed, the definition did not specify whether the

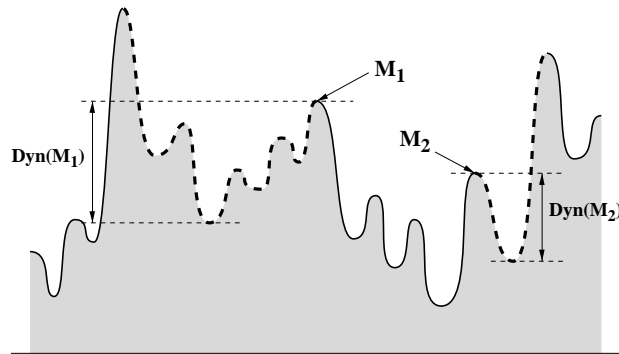


Figure 1.37 *Dynamics of maxima.*

maxima higher than m were (a) strictly higher, or (b) higher or equal. Neither is the correct answer, and explaining why would be beyond the scope of this chapter (this issue is discussed at length in). The important thing to remember is that Definition. 31 is not complete, and choices will have to be made in the implementation.

The dynamics provides a new and robust way to select maxima/minima based purely on their contrast—size and shape are irrelevant. For example, the task of extracting markers of white blobs in an image can be tackled using regional maxima. However, even if the image is only slightly noisy, it is more than likely that the blobs will exhibit multiple maxima. This can be addressed via dynamics: indeed, as illustrated by Fig. 1.38, only *one* of these maxima will have a significant dynamics value, whereas all the others will have low dynamics.

This principle can be put to good use with our electrophoresis gel image: in order to extract blob markers, the method used in Section 1.3.4 consisted of smoothing the image by alternating sequential filtering, and then extracting minima. An alternative method is simply to extract the minima whose dynamics is greater than or equal to a given threshold. This is illustrated in Fig. 1.39, and provide a set of markers that is at least as good as the one used previously.

The dynamics is one of the most useful tools recently added to the morphological toolbox. Just like the distance function can be seen as a multiscale erosion, and the granulometry function (see Section 1.4) can be seen as a multiscale opening, the dynamics is

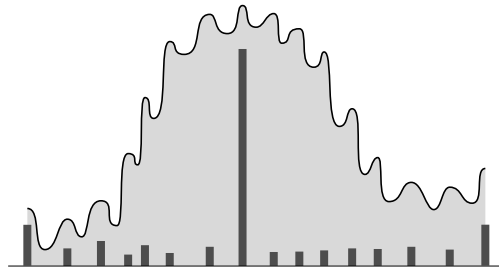


Figure 1.38 *Dynamics of the maxima inside a white blob: only one maximum has a significant dynamics value.*

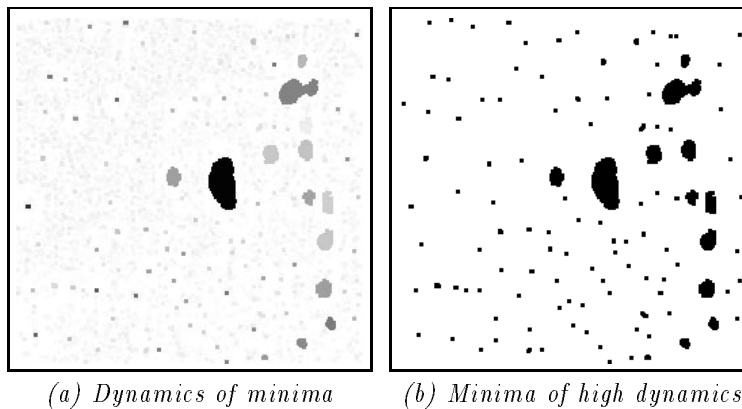


Figure 1.39 *Using the dynamics of minima to extract robust spot markers.*

the multiscale version of the operation consisting of subtracting a constant h from a gray image I , and reconstructing I from $I-h$. As such, it contains a lot of information that is invaluable in numerous segmentation application.

Fig. 1.40 illustrates this last point: the dynamics is the key element of the method that was designed to robustly separate touching grains in an automated inspection system. The actual segmentation algorithm designed to solve this particular application is a combination of both binary and grayscale segmentation techniques presented in Section 1.3.4.

To conclude this section, let us mention that the notion of dy-

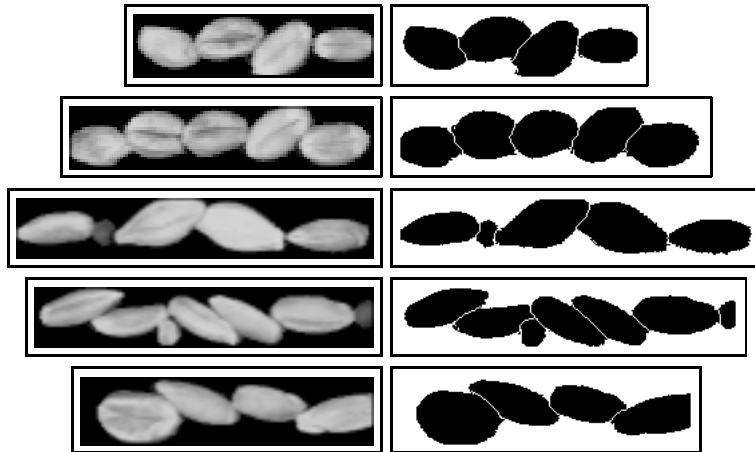


Figure 1.40 *Separation of touching grains in an automatic inspection system.*

namics extends to watershed lines in the following way: given an image I , and given h and h' , $h < h'$, we can extract the set S_h of minima of dynamics greater than or equal to h , and the set $S_{h'}$ of minima of dynamics greater than or equal to h' . Obviously:

$$S_{h'} \subseteq S_h.$$

Now, using S_h and $S_{h'}$ as sets of markers, we can use the method proposed in the Section 1.3.4 and extract the corresponding *marker-constrained* watershed lines of I , denoted by W_h and $W_{h'}$ respectively. Here again:

$$W_{h'} \subseteq W_h.$$

Indeed, a watershed line L located between a minimum m such that $m \in S_h, m \notin S_{h'}$, and any other minimum of S_h exists in W_h , but not in $W_{h'}$. From this, a method for tagging each watershed line with its dynamics is derived: the dynamics of a watershed line L is equal to the dynamics value h such that L exists in W_h , but not in W_{h+1} . This was originally proposed by Najman and Schmitt (1994).

As an illustration, consider the metallic alloy image of Fig. 1.21a. Using as markers the *maxima* of increasing dynamics, and extracting marker-driven watersheds on the *inverted* original image (to turn grain boundaries into crest-lines), a series of watershed images

is derived, each image containing the next one. The information contained in this series of images can be condensed into one image, namely the image of the dynamics-tagged watershed lines, also known as hierarchical watershed image. This is illustrated by Fig. 1.41.

Hierarchical watersheds constitute a very new concept, which the morphological image analysis community is just beginning to explore. Their usefulness for interactive segmentation (i.e., where the dynamics threshold is manually adjusted by a user) is already very clear. Hierarchical watersheds are also potentially very powerful for complex segmentation tasks. Furthermore, beyond dynamics, other criteria can be used to tag watershed edges, as described by Vachier and Vincent (1995). This leads to a large class of new watershed-based tools, and we can expect to read more and more about them, as image analysts begin to use them on practical applications and realize how powerful these techniques are.

1.4 Granulometries: Applications and Algorithms

1.4.1 Introduction, Background on Granulometries

In the present section, another major topic in morphological image analysis is covered: *granulometries*. They constitute one of the most useful and versatile sets of tools of morphological image analysis, and can be applied to a wide range of tasks, from feature extraction, to texture characterization, to size estimation, to image segmentation, etc. Though these operations are nearly 30 years old, they have become more popular and more useful in recent years, due to the general increase in computing power available at one's desktop, and due to recent advances in algorithms. These two factors combined make it possible to efficiently compute a whole range of granulometries, and this has led to new uses for these old tools.

Even today, traditional granulometry algorithms, which involve sequences of openings or closings with structuring elements of increasing size, are often prohibitively costly on non-specialized hardware. The primary goal of this section is to cover the most recent algorithmic advances. The proposed algorithms are often orders of magnitude faster than previously available techniques, thereby opening a range of new applications for granulometries. Several such applications of granulometries will be described in this section.

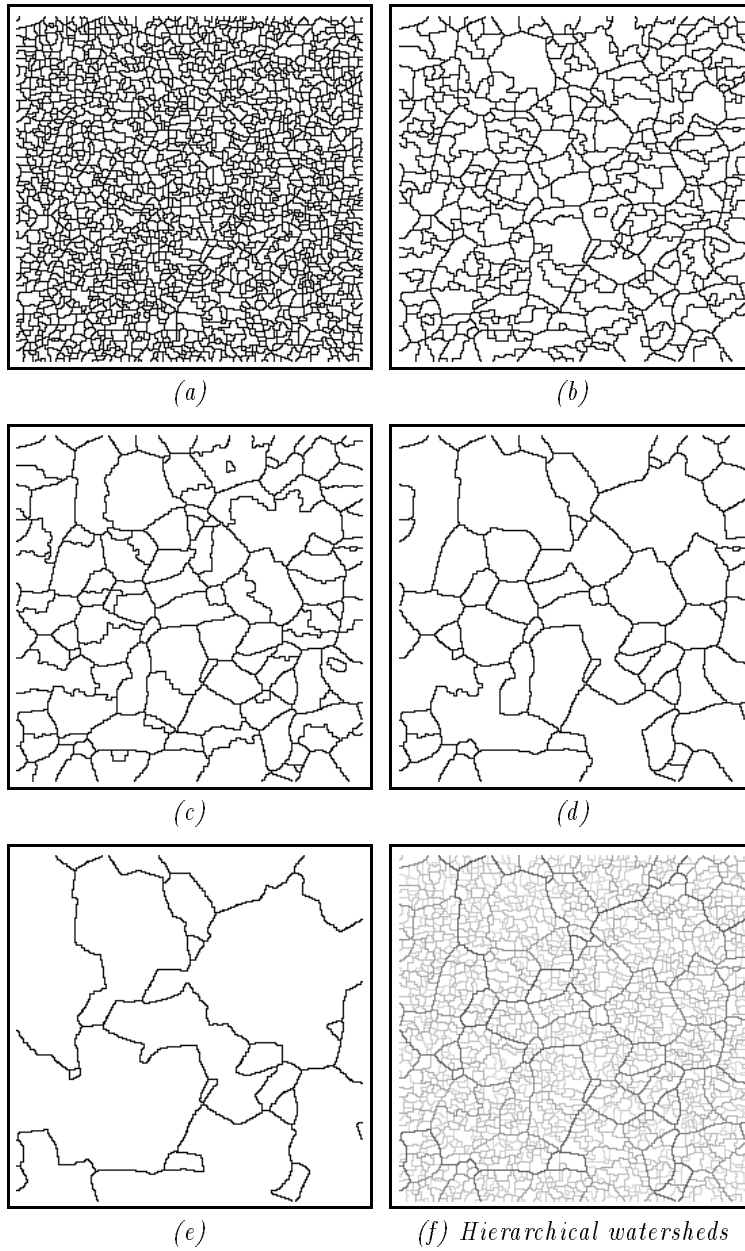


Figure 1.41 (a)–(e): watershed lines of Fig. 1.21a (inverted) when the set of markers used are maxima of increasing dynamics. (f): corresponding image of dynamics-tagged watershed lines, or hierarchical watersheds.

The concept of granulometries was introduced by Matheron (1967) as a new tool for studying porous media. The size of the pores in such media was characterized using series of openings with structuring elements of increasing size (Serra 1982). The theoretical study of these operations led Matheron to propose the following definition:

Definition 32 *Let $\Phi = (\phi_\lambda)_{\lambda \geq 0}$ be a family of image transformations depending on a unique parameter λ . This family constitutes a granulometry if and only if the following properties are satisfied:*

$$\forall \lambda \geq 0, \quad \phi_\lambda \text{ is increasing,} \quad (1.54)$$

$$\forall \lambda \geq 0, \quad \phi_\lambda \text{ is anti-extensive,} \quad (1.55)$$

$$\forall \lambda \geq 0, \mu \geq 0, \quad \phi_\lambda \phi_\mu = \phi_\mu \phi_\lambda = \phi_{\max(\lambda, \mu)} \quad (1.56)$$

Property (1.56) implies that for every $\lambda \geq 0$, ϕ_λ is an idempotent transformation. Therefore, $(\phi_\lambda)_{\lambda \geq 0}$ is nothing but a decreasing family of algebraic openings (Serra 1982). Conversely, one can prove that for any convex set B , the family of the openings with respect to $\lambda B = \{\lambda b \mid b \in B\}$, $\lambda \geq 0$, constitutes a granulometry (Matheron 1975).

More intuitively, suppose now that the transformations considered are acting on discrete binary images, or sets. In this context, a granulometry is a sequence of openings ϕ_n , indexed on an integer $n \geq 0$. Each opening is smaller than the previous one:

$$\forall X, \quad \forall n \geq m \geq 0, \quad \phi_n(X) \subseteq \phi_m(X). \quad (1.57)$$

The granulometric analysis of X with family of openings $(\phi_n)_{n \geq 0}$ is often compared to a *sifting* process: X is sifted through a series of sieves with increasing mesh size. Each opening (corresponding to one mesh size) removes more than the previous one, until the empty set is finally reached. The rate at which X is sifted is characteristic of this set and provides a “signature” of X with respect to the granulometry used. Denote by $m(A)$ the measure of a set A (area or number of pixels in 2-D, volume in 3-D, etc):

Definition 33 *The granulometric curve or pattern spectrum (Maragos 1989) of a set X with respect to a granulometry $\Phi = (\phi_n)_{n \geq 0}$ is the mapping $PS_\Phi(X)$ given by:*

$$PS_\Phi(X)(n) = m(\phi_n(X)) - m(\phi_{n-1}(X)). \quad (1.58)$$

Since $(\phi_n(X))_{n \geq 0}$ is a decreasing sequence of sets ($\phi_0(X) \supseteq \phi_1(X) \subseteq \phi_2(X) \supseteq \dots$), it is possible to condense its representation

by introducing the concept of granulometry function (Laÿ 1987, Vincent and Beucher 1989, Haralick, Chen and Kanungo 1992):

Definition 34 *The granulometry function or opening function $G_{\Phi}(X)$ of a binary image X for granulometry $\Phi = (\phi_n)_{n \geq 0}$ maps each pixel $x \in X$ to the size of the first n such that $x \notin \phi_n(X)$:*

$$G_{\Phi}(X)(x) = \min\{n > 0 \mid x \notin \phi_n(X)\}. \quad (1.59)$$

For any $n > 0$, the threshold of $G_{\Phi}(X)$ above a value n is equal to $\phi_n(X)$:

$$\phi_n(X) = \{p \in X \mid G_{\Phi}(X)(p) > n\}.$$

The following property follows immediately and states that the pattern spectrum can be obtained as histogram of the granulometry function:

Proposition 35 *The pattern spectrum $PS_{\Phi}(X)$ of X for granulometry $\Phi = (\phi_n)_{n \geq 0}$ can be derived from the granulometry function $G_{\Phi}(X)$ as follows:*

$$PS_{\Phi}(X)(n) = \text{card}\{p \mid G_{\Phi}(X)(p) = n\}, \quad (1.60)$$

where *card* stands for the cardinal (number of pixels) in a set.

This result is straightforward to prove. The concept of granulometry function is central to the algorithms described in section 1.4.4. An example of square granulometry function is shown in Fig. 1.42.

The granulometries that have been described so far are often referred to as granulometries *by openings*. By duality, granulometries *by closing* can also be defined; the granulometric analysis of a set X with respect to a family of closings is strictly equivalent to the granulometric analysis of X^C (complement of X) with the family of dual openings. Therefore, from now on, only granulometries by openings are considered. Similarly, these notions can be directly extended to grayscale images; in this context, the measure m chosen is the “volume” of the image processed, i.e. the sum of all its pixel values.

The granulometric analysis of Fig. 1.3a with respect to a family of openings with squares (as was used for Fig. 1.42) is shown in Fig. 1.43. From the resulting pattern spectrum, the dominant size of the beans (“size” being defined as size of the largest square a bean can contain) in this image can easily be derived. Granulometries therefore allow one to extract size information without any need for prior segmentation: the beans in this image are highly

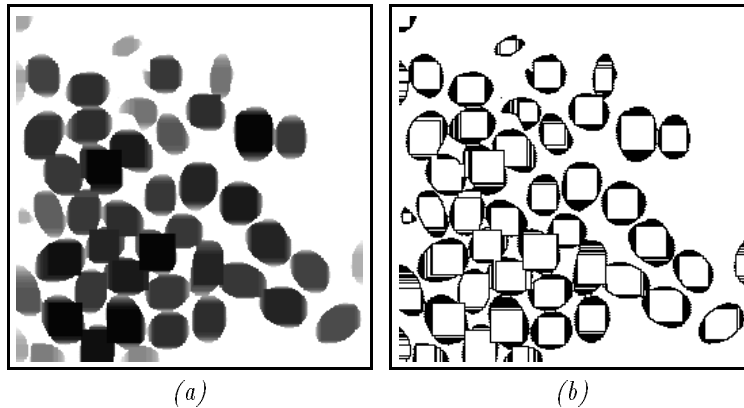
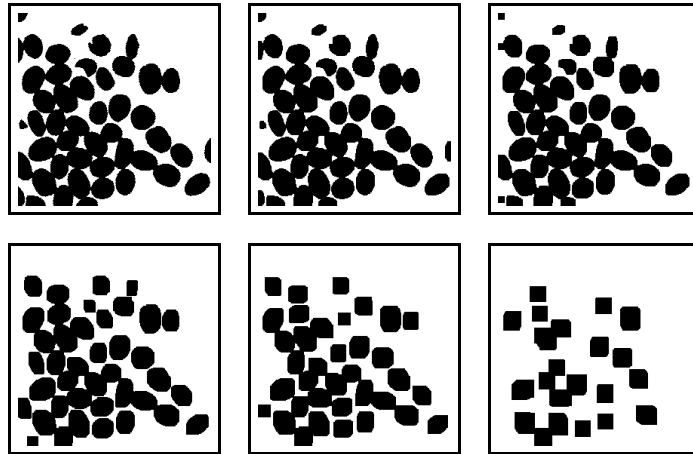


Figure 1.42 (a) Square granulometry function of coffee beans image of Fig. 1.3a, in which dark regions correspond to higher pixel value; (b) level lines of this granulometry function.

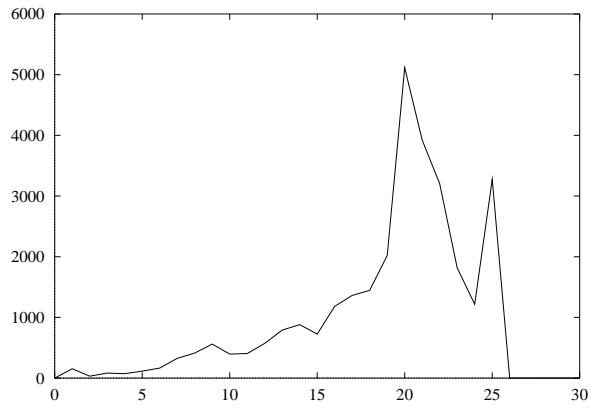
overlapping, yet their size can be estimated without individually identifying each bean.

Granulometries have been used for a variety of other image analysis tasks, including shape characterization and feature extraction (see for example Yang and Maragos 1993), texture classification (Chen and Dougherty 1992), and even segmentation (Dougherty *et al* 1992). Nonetheless, until recently, granulometric analysis involved performing a series of openings and/or closings of increasing size, which is prohibitively expensive for most applications, unless dedicated hardware is used.

In section 1.4.2, the literature on granulometry algorithms is briefly reviewed. A comprehensive set of fast algorithms for computing granulometries in binary images is then proposed: linear granulometries (i.e., granulometries based on openings with line segments) form the easiest case, and are computed using image “run-length”. They constitute the topic of Section 1.4.3. The 2-D case (granulometries with square or “diamond”-shaped structuring elements, or granulometries with unions of line-segments at different orientations) is covered in Section 1.4.4. It involves the determination of opening functions, and following Proposition 35, pattern spectra are then derived by simple histogramming. The grayscale case is then addressed in Section 1.4.5, and a very efficient algorithm for grayscale linear granulometries is described.



(a)



(b)

Figure 1.43 (a) Successive openings of a Fig. 1.3a using squares of increasing size as structuring elements. (b) Corresponding granulometric curve, or pattern spectrum: the peak at size 20 indicates the dominant size of the beans in the original binary image.

Finally, the concept of an *opening tree* is proposed in Section 1.4.6, and used to efficiently compute granulometries based on maxima of linear openings, minima of linear closings, and even “pseudo-granulometries” by minima of linear openings or maxima of linear closings. The efficiency of these latter algorithms enables one to use granulometries where previously unthinkable, a point which we illustrate on a number of examples.

1.4.2 Background on Granulometry Algorithms

The literature on mathematical morphology is not short of algorithms for computing erosions and dilations, openings and closings, with various structuring elements, in binary and in grayscale images. Reviewing them would be beyond the scope of this chapter. But no matter how efficient an opening algorithm is used, determining a pattern spectrum using a sequence of openings is a very time-consuming task given the number of operations involved. Furthermore, since the size of the structuring element increases with n , so does the computation time of the corresponding opening. Even if we assume that the computation time of $\phi_n(X)$ (n -th opening in the series) can be done in constant time (which is not always true depending on the structuring element and on the opening algorithm used), determining the pattern spectrum up to size n using openings is still an $O(n)$ algorithm.

The few granulometry algorithms found in literature only deal with the binary case, and have in common the use of granulometry functions as an intermediate step. The algorithm proposed by Yuan (1991) for determining binary square granulometries consists of first determining the *quench function* of the original set X . The quench function maps each pixel p of the skeleton (medial axis) S_X of X to the size (radius) $S_X(p)$ of the corresponding maximal square. (See Vincent (1991b) for more details on these concepts.) In a second step, each pixel p of the skeleton is replaced by a square centered at this pixel, with size $S_X(p)$, and gray-level $S_X(p) + 1$. The pixelwise maximum of all these squares provides the granulometry function of X . This algorithm is faster than the brute force method described in the previous paragraph, but still requires a significant amount of image scans. In addition, the more complicated the image or the larger the objects in it, the longer this method takes.

Surprisingly, a better algorithm can be found in an earlier pa-

per by Laÿ (1987), in which the author devotes a few lines to the description of a sequential algorithm (Rosenfeld and Pfaltz 1966, Vincent 1991a) based on the distance function (Rosenfeld and Pfaltz 1968, Borgefors 1986), and also using the granulometry function as an intermediate step. This algorithm still provides one of the most efficient implementations to date for binary granulometries with structuring elements such as squares and hexagons. In section 1.4.4, this technique is described in detail, and is extended to other types of binary granulometries.

The algorithm proposed by Haralick, Chen and Kanungo (1992) is interesting in that it allows in principle to compute granulometry functions with respect to any family of homothetic elements[§]. However, for simple structuring elements such as squares, this technique is not as efficient as the one mentioned in the previous paragraph, because its elementary steps (propagation and merging of lists of “propagators”) are rather computationally intensive, therefore slow.

1.4.3 Linear Granulometries in Binary Images

Linear granulometries in binary images constitute the simplest possible case of granulometries. Let us for example consider the horizontal granulometry, i.e., the granulometry by openings with the $(L_n)_{n \geq 0}$ family of structuring elements, where:

$$L_n = \underbrace{\bullet \bullet \bullet \cdots \bullet \bullet}_{n+1 \text{ pixels}} \quad (1.61)$$

From now on, we use the convention that the center of a structuring element is marked using a thicker dot than is used for the other pixels. Note that the location of the center of the structuring elements used has no influence on the resulting granulometry.

Let us analyze the effect of an opening by $L_n, n \geq 0$ on a discrete set X (binary image). The following notations are used from now on: the neighbors of a given pixel p in the square grid are denoted $N_0(p), N_1(p), \dots, N_7(p)$, and the eight elementary directions are encoded in the following way:

$$\begin{array}{ccc} 3 & 2 & 1 \\ & 4 & \bullet & 0 \\ 5 & 6 & 7 \end{array}$$

[§] The base element does not even have to be convex!

For a direction $d \in \{0, 1, \dots, 7\}$ and $k \geq 0$, we denote by $N_d^{(k)}(p)$ the k -th order neighbor of pixel p in direction d :

$$\begin{aligned} N_d^{(0)}(p) &= p, \quad \text{and } k > 0 \implies \\ N_d^{(k)}(p) &= N_d(N_d^{(k-1)}(p)). \end{aligned} \quad (1.62)$$

The opposite of direction d is denoted \check{d} . For example, if $d = 3$, then $\check{d} = 7$.

Definition 36 *The ray in direction d at pixel p in set X is given by:*

$$\begin{aligned} r_{X,d}(p) &= \{N_d^{(k)}(p) \mid k \geq 0 \text{ and} \\ &\forall 0 \leq j \leq k, N_d^{(j)}(p) \in X\}. \end{aligned} \quad (1.63)$$

With each pixel $p \in X$, we also associate a *run* in direction d , defined as the union of the rays in direction d and in direction \check{d} .

Definition 37 *The run in direction d at pixel p in set X is given by:*

$$R_{X,d}(p) = r_{X,d}(p) \cup r_{X,\check{d}}(p). \quad (1.64)$$

The number of pixels in a run R is called the *length* of this run and denoted $l(R)$.

The following proposition is immediate:

Proposition 38 *The opening of X by L_n , denoted here $X \circ L_n$, is the union of the horizontal runs $R_{X,0}(p)$ whose length is strictly greater than n :*

$$X \circ L_n = \bigcup_{p \in X} \{R_{X,0}(p) \mid l(R_{X,0}(p)) > n\}. \quad (1.65)$$

Therefore, any horizontal run of length n is left unchanged by all the openings with L_k , $k < n$, and is removed by any opening with L_k , $k \geq n$. Hence, the corresponding pattern spectrum PS_0 satisfies:

$$\text{PS}_0(X)(n) = \text{card}\{p \in X \mid l(R_{X,0}(p)) = n\}. \quad (1.66)$$

An extremely efficient 1-scan horizontal granulometry algorithm is easily derived from this formula:

Algorithm: horizontal binary granulometry

- Initialize pattern spectrum: for each $n > 0$, $\text{PS}[n] \leftarrow 0$
- Scan each line of image from left to right.
- In this process, each time a run R is discovered, do:

$$\text{PS}[l(R)] \leftarrow \text{PS}[l(R)] + l(R);$$

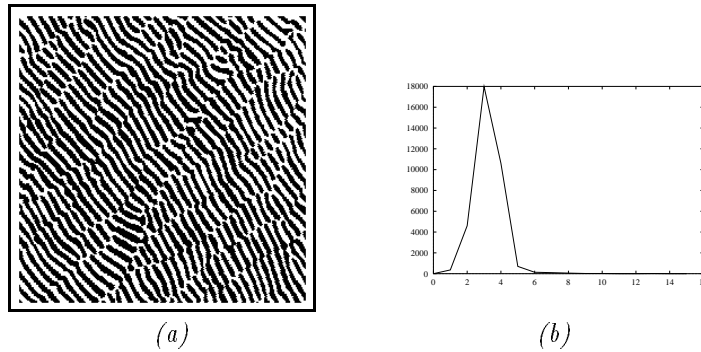


Figure 1.44 *Binary image of lamellar eutectics (a) and its granulometric curve using line segments at +45 degrees orientation (b).*

In applications where directional information is of interest, this algorithm provides a very useful and efficient way to extract size information characterizing the image under study. Consider for example Fig. 1.44a, which is a binary image of lamellar eutectics; Schmitt (1991) proposed a variety of methods for extracting the defect lines present in this image. Different methods used different kind of information about this image, and some required a knowledge of the typical width of the lamellae. This width can be accurately estimated by adapting the previous algorithm to the computation of linear granulometries at +45 degree orientation (direction perpendicular to the lamellae). The resulting pattern spectrum is shown in Fig. 1.44b, and its peak at index 3 indicates that the typical width of the lamellae is of 3 pixels.

1.4.4 Granulometry Functions on Binary Images

For non 1-D granulometries, the direct approach described in the previous section becomes intractable. Consider for example the case of a granulometry $(\phi_n)_{n \geq 0}$ where ϕ_n is a maximum of openings with the horizontal segment L_n and its vertical counterpart L_n^\perp . For each pixel, it becomes necessary to know the size of the horizontal run as well as the vertical run it belongs to.

Linear granulometry functions are therefore the required step. Given the horizontal and the vertical granulometry functions of X , the granulometry function of X corresponding to the $(\phi_n)_{n \geq 0}$ of previous paragraph is simply obtained as a pixelwise maximum.

More generally, the same is true for any two granulometry functions, and the following proposition can be stated:

Proposition 39 *Let $\Phi = (\phi_n)_{n \geq 0}$ and $\Psi = (\psi_n)_{n \geq 0}$ be two granulometries. Then, $\max(\Phi, \Psi) = (\max(\phi_n, \psi_n))_{n \geq 0}$ is also a granulometry and for any set X :*

$$G_{\max(\Phi, \Psi)}(X) = \max(G_\phi(X), G_\psi(X)). \quad (1.67)$$

Indeed, for any n , $\max(\phi_n, \psi_n)$ is an opening, and for any $n, m, n \leq m$, $\max(\phi_n, \psi_n) \geq \max(\phi_m, \psi_m)$. Moreover, for any $x \in X$:

$$\begin{aligned} G_{\max(\Phi, \Psi)}(X)(x) &= \min\{n > 0 \mid x \notin (\max(\phi_n, \psi_n))(X)\} \\ &= 1 + \max\{n > 0 \mid x \in (\max(\phi_n, \psi_n))(X)\} \\ &= 1 + \max\{n > 0 \mid x \in \phi_n(X) \cup \psi_n(X)\} \\ &= \max\{1 + \max\{n > 0 \mid x \in \phi_n(X)\}, \\ &\quad 1 + \max\{n > 0 \mid x \in \psi_n(X)\}\} \\ &= \max\{G_\phi(X)(x), G_\psi(X)(x)\} \end{aligned}$$

which completes the proof. Determining the linear granulometry function of a binary image is a relatively straightforward task. Take for example the horizontal case: like in the previous section, the principle of the granulometry function algorithm is to locate each horizontal run. But now, in addition, each run R gets also tagged with its length $l(R)$. This involves scanning the black (ON) pixels of the image twice, and the white (OFF) pixels only once. The resulting algorithm is straightforward and is hardly more time consuming than the one described in Section 1.4.3. Examples of linear granulometry functions are shown in Fig. 1.45 and in Fig. 1.46.

The case of truly 2-D binary granulometry functions is the next level up in complexity. In the rest of this section, we focus on granulometry functions $G_S(X)$ based on openings with the homothetics of elementary square S . The algorithm described extends to granulometry functions $G_D(X)$, based on openings with the homothetics of “diamond” shape D , as described by Vincent (1994a).

$$S = \begin{array}{cc} & \bullet \\ \bullet & \bullet \\ \bullet & \bullet \end{array} ; \quad D = \begin{array}{ccc} & \bullet & \\ \bullet & \bullet & \bullet \\ & \bullet & \end{array} \quad (1.68)$$

Together with the linear case, these granulometries cover 99% of all practical needs.

Like in Haralick’s algorithm (Haralick, Chen and Kanungo

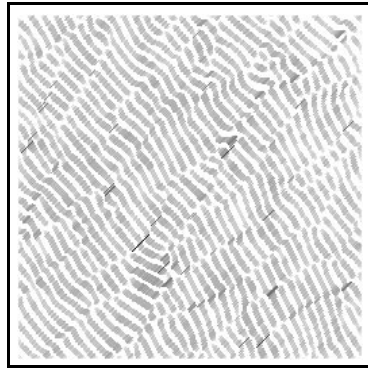


Figure 1.45 *Linear granulometry function of image 1.44a, whose histogram directly provides the curve of Fig. 1.44b.*

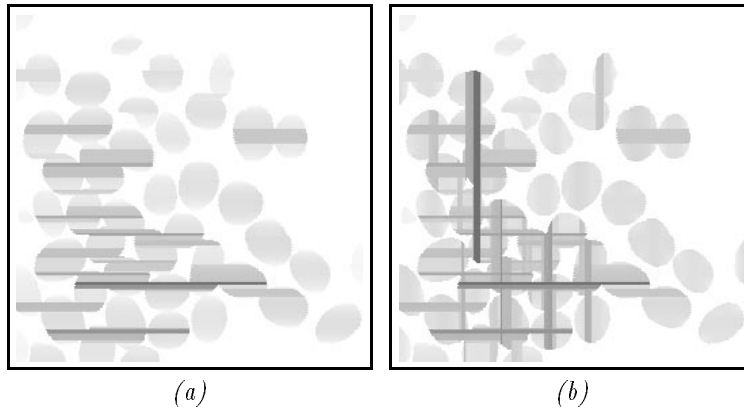


Figure 1.46 (a) *Horizontal granulometry function.* (b) *Pixelwise maximum of horizontal and vertical granulometry functions provides the granulometry function corresponding to maxima of openings with vertical and horizontal line segments.*

1992), the first step of the proposed technique consists of computing what some authors have called a *generalized distance function* (Bertrand and Wang 1988, Haralick and Shapiro 1991). Let B be an arbitrary structuring element containing its center. Let

$$nB = \underbrace{B \oplus B \oplus \dots \oplus B}_{n \text{ times}}$$

denote the structuring element “of size n ”. Let also ε_B denote the erosion by structuring element B (Serra 1982):

$$\varepsilon_B(X) = X \ominus \check{B}.$$

Definition 40 *The generalized distance function $d_B(X)$ with respect to the family of structuring elements $(nB)_{n>0}$ assigns to each pixel $p \in X$ the smallest $k > 0$ such that $p \notin \varepsilon_{kB}(X)$:*

$$d_B(X)(p) = \min\{k > 0 \mid p \notin \varepsilon_{kB}(X)\}. \quad (1.69)$$

Generalized distance functions are determined using sequential algorithms that are straightforwardly derived from the original algorithm proposed by Rosenfeld and Pfaltz (1966, 1968). When the center of the structuring element is in the bottom-right corner of element B (last pixel met in a raster-order scan of this element), the distance function $d_B(X)$ can be computed in one single raster scan.

In the case where $B = S$ (see Eq. (1.68)), the following algorithm can be proposed:

Algorithm: Generalized dist. func. with square S

- **Input:** binary image I of set X
- **Scan I in raster order;**
 - Let p be the current pixel;
 - if $I(p) = 1$ (p is in X):

$$I(p) \leftarrow \min\{I(N_4(p)), I(N_3(p)), I(N_2(p))\} + 1;$$

An example of generalized distance function resulting from this algorithm is shown in Figs. 1.47a–b. A way to interpret the result is to say that, for each pixel p , if one was to translate structuring element $[d_S(X)(p)]S$ so that its center coincides with p , this translated element—denoted $p + [d_S(X)(p)]S$ —would be entirely included in X . However, $p + [d_S(X)(p) + 1]S \notin X$. We can therefore state the following proposition:

Proposition 41 *The granulometry function $G_S(X)$ is obtained from $d_S(X)$ as follows:*

$$\forall p \in X, \quad G_S(X)(p) = \max\{d_S(X)(q) \mid p \in (q + d_S(X)(q)S)\}. \quad (1.70)$$

This result is intuitively obvious and can be proved easily. In algorithmic terms, it means that we can compute $G_S(X)$ by propagating the value $d_S(X)(p)$ of each pixel p over the square

$p + d_S(X)(p)S$, and then by taking the pixelwise maximum of the values propagated at each pixel.

In the technique proposed by Haralick, Chen and Kanungo (1992), this propagation step is achieved via an anti-raster scan of the distance function image, in which, at each pixel, a list of propagated values is maintained. In the particular case of square granulometry function $G_S(X)$, computing the value at pixel p as well as the list of propagated values at p , requires a merging of the lists of propagated values at pixels $N_0(p)$, $N_6(p)$, and $N_7(p)$.

This merging step turns out to be expensive, and in the case of square granulometry function $G_S(X)$, a less general, but much more efficient technique can be proposed. This technique takes advantage of the fact that square S can be decomposed into the Minkowski addition of the two elementary line segments E_1 and E_2 :

$$S = \begin{matrix} \bullet & \bullet \\ \bullet & \bullet \end{matrix} = \begin{matrix} \bullet & \bullet \\ \bullet & \bullet \end{matrix} \oplus \begin{matrix} \bullet \\ \bullet \end{matrix} = E_1 \oplus E_2 \quad (1.71)$$

Therefore, the complex propagation step of the granulometry function algorithm described by Haralick, Chen and Kanungo (1992) can in fact be decomposed into two much simpler propagations, with substantial speed gain. The distance function extraction step is followed by two linear propagation steps that are identical, except that one propagates distance values leftward in each line, whereas the other one propagates values upward in each column.

The algorithm for right-to-left propagation of distance values is given below. Its principle is to propagate each pixel value $I(p)$ to the left $I(p) - 1$ times, or until a larger value v is found, in which case the list of propagated values is reset to this value. . . The algorithm maintains an array **propag** containing the number of times each value remains to be propagated.

Algorithm: Left propag. of dist. values of $d_S(X)$

- Input: image I of the generalized distance function $d_S(X)$;
- For each line of the image, do:
 - Initializations: $\text{maxval} \leftarrow 0$ (current maximal value propagated);
 - Scan line from right to left:
 - Let p be the current pixel;
 - If $I(p) \neq 0$:
 - If $I(p) > \text{maxval}$:
 - $\text{maxval} \leftarrow I(p)$;

```

propag[I(p)] ← I(p);
∀0 < i ≤ maxval, propag[i] ← propag[i] - 1;
maxval ← largest i ≤ maxval with propag[i] ≥ 0;
I(p) ← maxval

```

A few implementation tricks can speed up computation by substantially reducing the number of times the entire array `propag` is scanned per scanline. Their description would be beyond the scope of this chapter (see for more a more in-depth discussion). The resulting algorithm is quasi-linear with respect to the number of pixels in the image, and is almost independent of object size (see Table. 1.1). Using again the coffee bean image as running example, the result of this propagation step is shown in Figs. 1.47c–d, and the final granulometry function obtained after upward propagation in each column is shown in Figs. 1.47e–f.

This algorithm can be adapted for granulometry functions with any structuring element that can be decomposed as a Minkowski addition of the elementary line segments E_1 , E_2 , E_3 , and E_4 (See Eqs. (1.71) and (1.72)). It also extends to the computation of hexagonal opening functions in the hexagonal grid (Laÿ 1987). Furthermore, although diamond D cannot be decomposed as the Minkowski addition of two line segments:

$$D = \begin{array}{c} \bullet \\ \bullet \bullet \bullet \\ \bullet \end{array} \neq \begin{array}{c} \bullet \oplus \bullet \\ \bullet \end{array} = E_3 \oplus E_4, \quad (1.72)$$

the algorithm can be adapted to the computation of granulometry functions based on openings with the homothetics of D , as described by Vincent (1994a).

Table 1.1 summarizes the speed of these granulometry functions on the 256×256 coffee bean image used as running example. We chose not to compare these timings with those of traditional opening-based algorithms. The speed of the latter algorithms can indeed vary tremendously depending on the quality of the implementation. Note however that for this coffee bean image, which has approximately 30000 black pixels, Haralick, Chen, and Kanungo's algorithm takes between 0.5s and 0.6s to compute the square granulometry function shown in Fig. 1.47e, on a Sparc Station 2. This workstation being between two and three times slower than a Sparc Station 10, we can conclude that the algorithm described here is between three and four times faster.

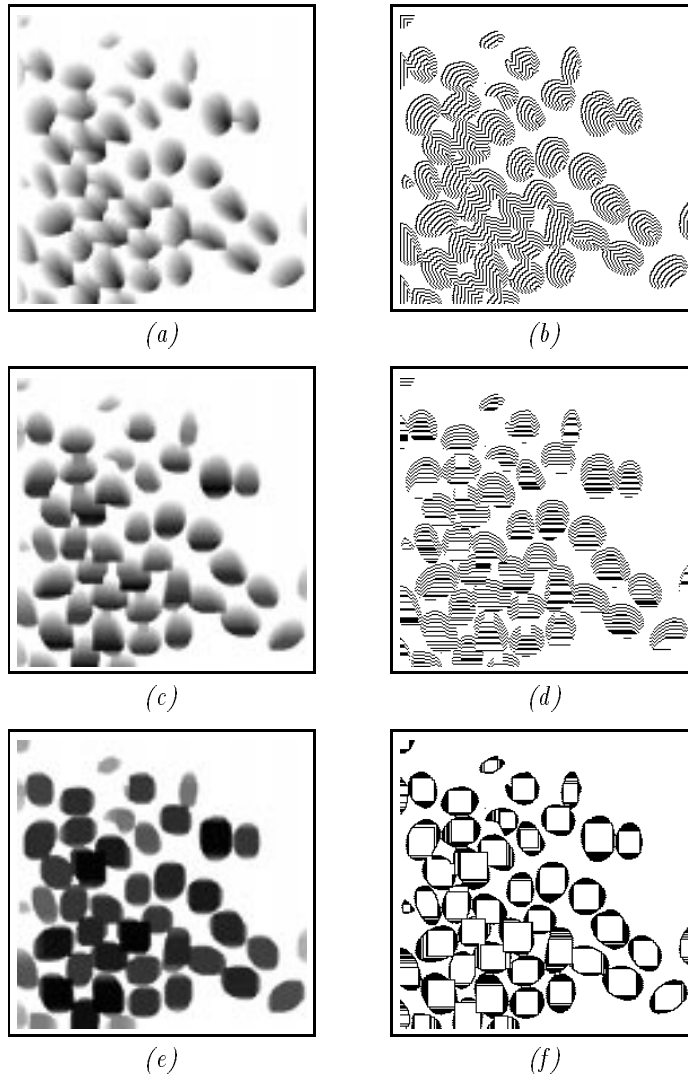


Figure 1.47 *Computation of granulometry function using square structuring elements. (a) generalized distance function; (b) level lines; (c) propagation of values from right to left; (d) level lines; (e) final granulometry function; (f) level lines of granulometry function.*

Table 1.1 *Execution time of various granulometry function algorithms on the 256×256 coffee bean image, measured on a Sun Sparc Station 10 workstation.*

Type of granulometry function	Execution time
horizontal	0.018s
max in 4 directions	0.207s
square	0.085s
“diamond”	0.094s

1.4.5 Linear Grayscale Granulometries

Grayscale granulometries are potentially even more useful than binary ones, because they enable the extraction of information directly from grayscale images. A number of theoretical results have been published on them (see e.g. Kraus *et al* 1992); however, since until recently, no efficient technique was available to compute grayscale granulometries, they have not been used very much in practice. In this section, we remedy this situation and describe a new algorithm for computing linear grayscale granulometries. In the next section, we extend it to granulometries based on maxima of linear openings, as well as pseudo square granulometries. For more details, refer to (Vincent 1994b).

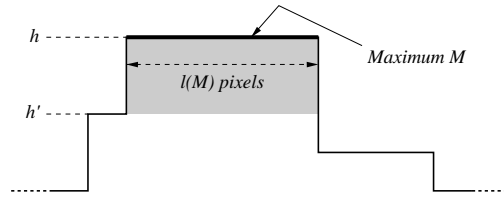
Without loss of generality, let us consider the horizontal case. The structuring elements considered now are the L_n 's of equation (1.61). Let I be a discrete grayscale image.

Definition 42 *A horizontal maximum M of length $l(M) = n$ in grayscale image I is a set of pixels $\{p, N_0^{(1)}(p), N_0^{(2)}(p), \dots, N_0^{(n-1)}(p)\}$ such that*

$$\forall i, 0 < i < n, I(N_0^{(i)}(p)) = I(p) \quad \text{and} \\ I(N_4(p)) < I(p), \quad I(N_0^{(n)}(p)) < I(p). \quad (1.73)$$

This notion is the 1-D equivalent of the classic *regional maximum* concept described in Section 1.3. The study of how such maxima are altered through horizontal openings is at the basis of the algorithm described here. The following proposition holds:

Proposition 43 *Let M be a horizontal maximum of I . Let $p_i \in M$*



and $p_r \in M$ respectively denote the extreme left pixel and the extreme right pixel of M . Let $n = l(M)$ be the length of this maximum. Then, for any $p \in M$:

$$\forall k < n, \quad (I \circ L_k)(p) = I(p), \quad (1.74)$$

$$\text{for } k = n, \quad (I \circ L_n)(p) = \max\{I(N_l(p_l)), I(N_r(p_r))\} < I(p), \quad (1.75)$$

$$\forall k > n, \quad (I \circ L_k)(p) < I(p). \quad (1.76)$$

The proof for this proposition is straightforward. Phrased differently, this result states that any opening of I by a line segment L_k such that $k < n$ leaves this maximum unchanged, whereas for any $k \geq n$, all the pixels of M have a lower value in $I \circ L_k$ than in I . Furthermore, we can quantify the effect of an opening of size n on the pixels of this maximum: the value of each pixel $p \in M$ is decreased from $I(p)$ to $\max\{I(N_l(p_l)), I(N_r(p_r))\}$. In granulometric terms, the contribution of maximum M to the n -th value of the horizontal pattern spectrum $PS_h(I)$ is:

$$n \times [I(p) - \max\{I(N_l(p_l)), I(N_r(p_r))\}]. \quad (1.77)$$

This is illustrated by Fig. 1.48.

Additionally, the local effect of a horizontal opening of size n on M is that a new “plateau” of pixels is created at altitude $\max\{I(N_l(p_l)), I(N_r(p_r))\}$. This plateau P contains M , and may be itself a maximum of $I \circ L_n$. If it is, we say that P is part of the maximal region $R(M)$ surrounding maximum M , and we can now compute the contribution of P to the $l(P)$ -th bin of the pattern spectrum, etc.

Following these remarks, the principle of the present grayscale granulometry algorithm is to scan the lines of I one after the other.

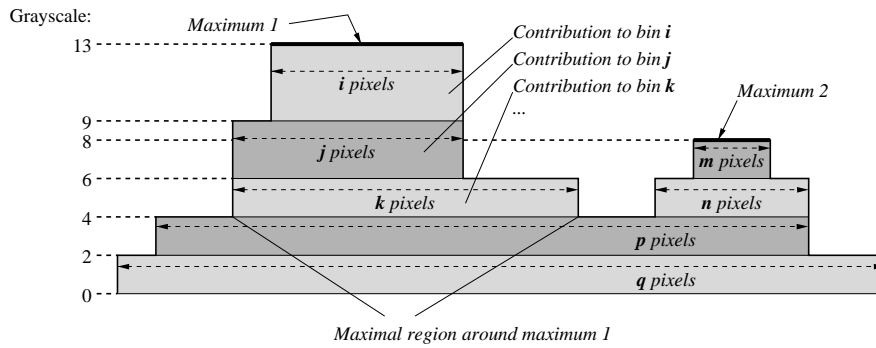


Figure 1.49 *Illustration of horizontal grayscale granulometry algorithm for a line with two maxima. In this case, maxima were scanned from left to right. Processing maximum 1 results in bins i , j , and k of the pattern spectrum being incremented. While processing maximum 2, bins m and n are first incremented, then the algorithm skips over the already processed maximal region, and bins p and q are incremented. The maximal region around maximum 2 is the entire line.*

Each horizontal maximum M of the current line is identified, and its contribution to $PS_0(I)(l(M))$ is determined. If it turns out that after opening of size $l(M)$, the new plateau formed is still a maximum, the contribution of this maximum to the pattern spectrum is computed as well. The process is iterated until the plateau formed by opening is no longer a maximum, or until it becomes equal to the entire scanline considered. The next maximum of the current line is then considered, etc. This process is illustrated by Fig. 1.49. Specifically, the algorithm works as follows on each image line:

Algorithm: horizontal granul. of an image line

- for each maximum M of this line (in any order) do:
 - Adds contribution of this maximum to $l(M)$ -th bin of pattern spectrum;
 - Let P be the plateau of pixels formed by opening of size $l(M)$ of M ; if P is itself a maximum of $I \circ L_{l(M)}$, computes its contribution to bin $l(P)$ of pattern spectrum;
 - Iterate previous step until the new plateau formed is no longer a maximum;
 - ‘Mark’ the maximal region around M as already processed;

Note that this algorithm is inherently recursive: once a maximal region R has been processed, all of its pixels are regarded as having the gray-level they were given by the last opening considered for R . In practice though, there is no need to physically modify the values of all the pixels in R : keeping track of the first and last pixels of R is sufficient, allowing the algorithm to efficiently skip over already processed maximal regions (see Fig. 1.49). Thanks to this trick, the algorithm only considers each image pixel *twice* in the worst case.

This algorithm was compared to the traditional opening-based technique. For the latter, a fast opening algorithm was used, whose speed is proportional to the number of pixels in the image and (almost) independent of the length of the line segment used as structuring element. As illustrated by table 1.2, the new algorithm described in this section is *three orders of magnitude faster*.

We compared the speed of this algorithm to the traditional opening-based technique. For this latter, a highly optimized opening algorithm was used, which is linear with respect to the number of pixels in the image, and whose speed is (almost) independent of the length of the line segment used as structuring element. Both original 512×512 weld images of Fig. 1.50 were used for this comparison. As illustrated by table 1.2, the algorithm described in this section is up to three orders of magnitude faster.

The speed of this new algorithm opens a range of new applications for grayscale granulometries. Traditionally, the practical problems granulometries have been used to address dealt with either texture discrimination, or feature extraction for object recognition. In the first case, either computation time was not an issue, or the discrimination task could be performed off-line. In the second case, granulometries were computed on very small images (e.g. characters), so that computation time could remain reasonable.

With this algorithm, it becomes possible to use grayscale granulometries more systematically: these tools indeed provide an efficient and accurate way to extract *global* size information directly from a grayscale image. Extracting this information is sometimes a goal in itself; but this size estimation can also be essential to calibrate the parameters of, e.g., an image segmentation algorithm, thereby greatly enhancing the robustness of this algorithm.

Figs. 1.50a–b are used to illustrate how grayscale granulometries

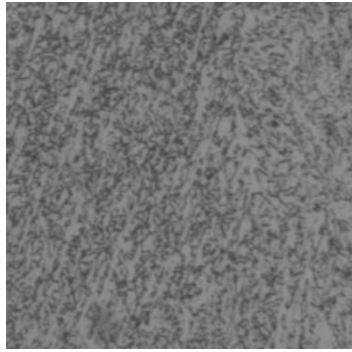
can be used to estimate size information[¶]. These figures represent welds at a high magnification. The quality of these welds is related to the size, shape, and organization of the light patterns observed in Figs. 1.50a–b. To estimate the size of the typical patterns in each image, linear granulometries were used, both in the vertical and in the horizontal direction. The resulting pattern spectra are shown in Figs. 1.50c–d. First, one can observe that the horizontal granulometric curve is very similar to the vertical one; we conclude that the patterns in images 1.50a and 1.50b do not have any preferential orientation. Second, the curves in Fig. 1.50c exhibit a well-marked peak for size 4, whereas the peak of the curves in Fig. 1.50d is found for size 12. We conclude that the typical width and height of the patterns in Figs. 1.50a and in Fig. 1.50b is of 4 pixels and 12 pixels respectively. Additionally, Fig. 1.50d shows the the distribution of pattern sizes in Fig. 1.50b is rather wide.

Another example of application is shown in Fig. 1.51. Images 1.51a and 1.51b respectively show X-rays of a healthy lung and of a lung exhibiting signs of the “miner’s disease”. These two images look very different from a texture point of view, though it would be hard for a segmentation algorithm to specifically extract the “objects” that make image b different from image a. Essentially, the former image shows a very smooth texture while the latter is scattered with white nodules, which makes the texture rougher. Grayscale granulometries are the instrument of choice to discriminate between these two types of textures. To that end, M. Grimaud, author of the original study, had used grayscale granulometries by openings and closings with squares. The resulting pattern spectra for both X-ray images, shown in Fig. 1.51a–b, are extremely different from one another. However, in order to effectively discriminate between these two images, expensive grayscale granulometries with squares are not necessary: using the algorithms presented here to extract *horizontal* grayscale granulometric curves is just as effective, as shown in Fig. 1.51d, and is orders of magnitude faster.

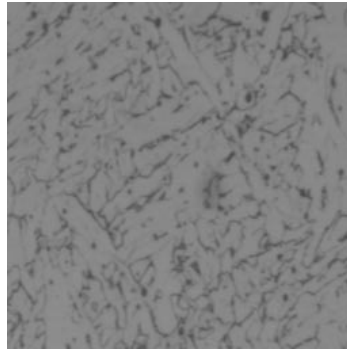
1.4.6 Granulometries with Maxima of Linear Openings

In order to deal with more complicated cases, we now describe a technique that can be seen as a generalization of the concept of opening functions for the grayscale case. When performing open-

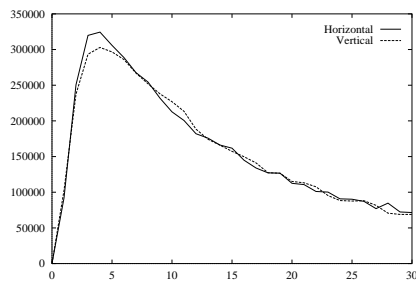
[¶] Images gracefully provided by DMS, CSIRO, Australia.



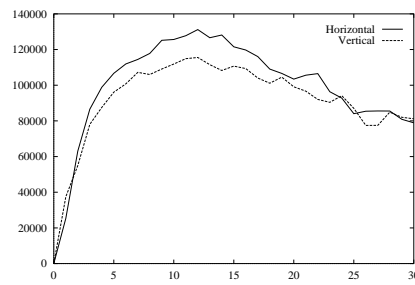
(a)



(b)



(c)

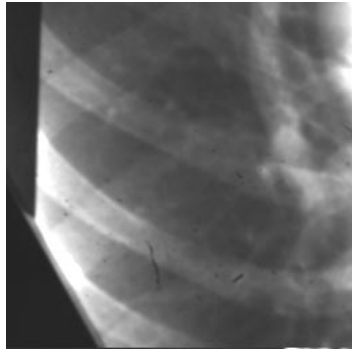


(d)

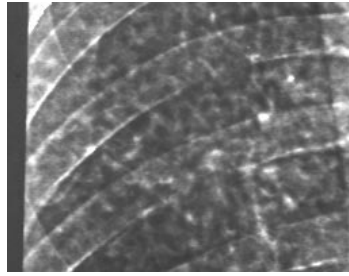
Figure 1.50 Using linear grayscale granulometries to estimate object size without prior segmentation. Curve (c) clearly indicates that the typical width and height of the white patterns in image (a) is 4 pixels. Similarly, curve (d) shows that the typical width/height of the patterns in (b) is 12.

ings of increasing size of a binary image, each “on” pixel p is turned “off” for an opening size given by the value of the opening function at pixel p . In other words, the opening function encodes for each pixel the successive values it takes for increasing opening sizes (namely, a series of 1’s followed by a series of 0’s). Similarly, in the grayscale case, as the size of the opening increases, the value of each pixel decreases monotonically. If each pixel was assigned the list of values it takes for every opening size, then the corresponding grayscale granulometry could be extracted straightforwardly.

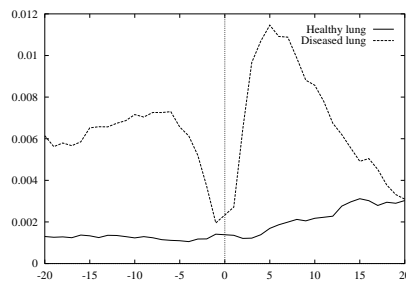
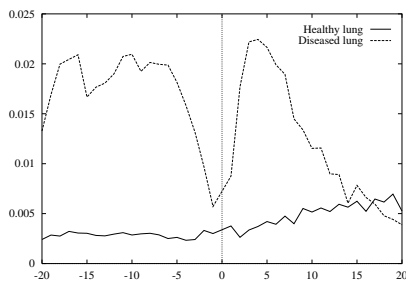
Unfortunately, even if it were possible to compute such lists of



(a) X-ray of healthy lung



(b) X-ray of diseased lung



(c) Square granulometric curves (d) Linear granulometric curves

Figure 1.51 *Example of texture differentiation using grayscale granulometries by openings and closings. The pattern spectra of (a) and (b) are completely different, whether linear or square granulometries are used (original study due to M. Grimaud).*

values quickly, assigning one to each image pixel would require far too much memory. A more compact representation needs to be designed, that takes into account the intrinsic “redundancy” of opened images, characterized by their large plateaus of pixels. If we again consider the linear case, an elegant solution can be proposed to both the problem of computing these lists of values, and the problem of storing them compactly:

Let M be a horizontal maximum of image I , with altitude (grayscale) h . We pointed out in the previous section that a horizontal opening of size $l(M)$ of M takes all of its pixels down to a

new value h' . Beyond this, the following proposition can easily be proved:

Proposition 44 *Let $n > 0$, I a grayscale image such that $I = I \circ L_{n-1}$. Then, for every pixel p in I :*

$$(I \circ L_n)(p) < I(p) \iff \exists M \text{ horiz maximum, } l(M) = n \text{ and } p \in M. \quad (1.78)$$

Therefore, at opening n in the sequence, the only pixels affected are those which belong to maxima of length n . Furthermore, all the pixels belonging to the same maximum M , $l(M) = n$, will be affected in the same way for any opening of size greater than or equal to n . The list of decreasing values we wish to associate with each pixel in M can therefore “converge” into one single list for size n . For larger opening sizes, this list may itself be merged with other lists, etc.

Based on this principle, the present algorithm represents each image line as a tree T , which we call its *opening tree*. The leaves of T are the image pixels, and the nodes are made of pairs (h, n) , where h is a grayscale value and n is an opening size. Every pixel that can be reached by going upwards in the tree starting from node (h, n) is such that its value for the opening of size n is h . Conversely, starting from a pixel p , successive pairs $(h_1, n_1), (h_2, n_2), \dots, (h_i, n_i), \dots$, are reached by going down towards the root of the tree. By convention, for this pixel, $(h_0, n_0) = (I(p), 0)$. These pairs satisfy:

$$\forall i > 0, \quad h_i > h_{i+1} \text{ and } n_i < n_{i+1}. \quad (1.79)$$

For $n \geq 0$, the value of the opening of size n of I at pixel p is given by:

$$(I \circ L_n)(p) = h_j, \quad \text{where } j \text{ is such that } n_j \leq n < n_{j+1}. \quad (1.80)$$

Opening trees can be computed using an algorithm very similar to the one described in the previous section. An example of opening tree is shown in Fig. 1.52.

Opening trees provide a hierarchical description that can be used to compactly represent *all* the horizontal openings of a grayscale image. In this respect, this notion is a grayscale equivalent of the opening function mentioned earlier (see Fig. 1.42). One can prove that, in the worst case, the opening tree has one node per image

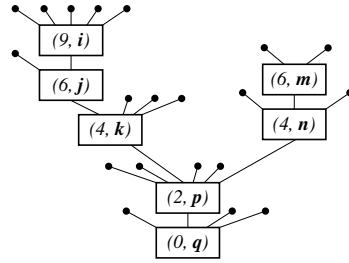


Figure 1.52 *Opening tree representation of the image line of Fig. 1.49. The leaves of this tree (\bullet) correspond to the image pixels.*

pixel. In practice though, only between 0.3 and 0.9 nodes per pixel are needed depending on the complexity of the image processed.

Any horizontal opening of I can be straightforwardly derived from its opening tree. In addition, the horizontal pattern spectrum of I , $PS_h(I)$, can be computed from T as follows:

Algorithm: horiz granul of I from its opening tree

- initialize each bin of pattern spectrum $PS_h(I)$ to 0;
- for each pixel p of I do:
 - $v \leftarrow I(p)$; $(h, n) \leftarrow$ node pointed at by p ;
 - while (h, n) exists, do:
 - $PS_h(I)(n) \leftarrow PS_h(I)(n) + (v - h)$;
 - $v \leftarrow h$; $(h, n) \leftarrow$ next node down in tree;

This algorithm is obviously less efficient for horizontal granulometries than the one described in the previous section. However, it easily generalizes to the computation of granulometries using maxima of linear openings in several orientations. For example, to determine the granulometric curve corresponding to maxima of horizontal and vertical openings, one first extracts the horizontal opening tree T_1 and the vertical opening tree T_2 . Then, for each pixel p , the technique consists of descending the corresponding branches of T_1 and T_2 simultaneously as follows:

Algorithm: granul of I from trees T_1 and T_2

- initialize each bin of pattern spectrum $PS(I)$ to 0;
- for each pixel p of I do:
 - $v \leftarrow I(p)$;
 - $(h_1, n_1) \leftarrow$ node of T_1 pointed at by p ;

```

-  $(h_2, n_2) \leftarrow$  node of  $T_2$  pointed at by  $p$ ;
- while  $(h_1, n_1)$  and  $(h_2, n_2)$  exist, do:
  size  $\leftarrow$   $\max(n_1, n_2)$ ;
  while  $n_1 \leq$  size and  $(h_1, n_1)$  exists do:
     $(h_1, n_1) \leftarrow$  next node down in  $T_1$ ;
  while  $n_2 \leq$  size and  $(h_2, n_2)$  exists do:
     $(h_2, n_2) \leftarrow$  next node down in  $T_2$ ;
   $\text{PS}(I)(\text{size}) \leftarrow \text{PS}(I)(\text{size}) + (v - \max(h_1, h_2))$ ;
   $v \leftarrow \max(h_1, h_2)$ ;

```

The same technique extends to any number of opening trees. The whole granulometry algorithm (extraction of trees followed by computation of the pattern spectrum from these trees) is once again orders of magnitude faster than traditional techniques, as illustrated by table 1.2. In this table, a granulometry by maxima of linear openings at 4 orientations was computed. In terms of memory, the computation of this particular granulometry requires, in the worst case, 1 pointer (4 bytes) and 1 node (8 bytes) per pixel, for each orientation. This comes to a total of 48 bytes/pixel, i.e., a worst case scenario of 12 Megabytes for a 512×512 image. This is a reasonable tradeoff given the speed of the algorithm, and is not a strain on modern systems.

Table 1.2 *Execution time of traditional opening-based techniques and of present algorithms for the computation of a horizontal grayscale granulometry (left) and a granulometry by maxima of linear openings in 4 orientations (right). A complex 512×512 image was used for this comparison, done on a Sun Sparc Station 10. Granulometries were computed for opening sizes 1 to 512.*

horizontal		
traditional	new	improvement factor
204s	0.206s	990
in 4 orientations		
traditional	new	improvement factor
824s	2.78s	296

The algorithm given above can be easily adjusted to the computation of *pseudo-granulometries* by *minima* of linear openings.

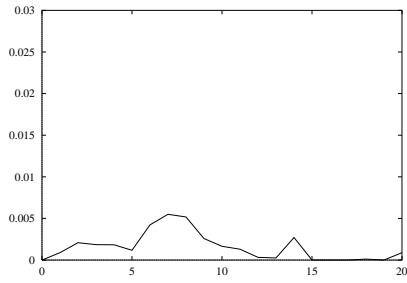
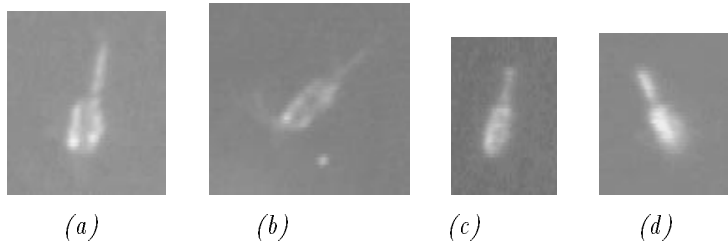
Although minima of openings are not themselves openings (Mathéron 1975), minima of openings with line segments of increasing length constitute a decreasing family of image operators. The resulting pseudo-granulometric curves often characterize the same image features as square granulometries.

Together with the linear granulometries described in this section, these pseudo square granulometries have been very successful as one of the feature sets used to characterize plankton in towed video microscopy images^{||} (Davis *et al* 1992, Tang *et al* 1996). As an illustration, Fig. 1.53 shows four different images of copepod oithona and their corresponding pattern spectra, for a granulometry using openings with squares. These curves are all relatively flat, exhibiting a single well-marked maximum, between size 5 and 10, which corresponds to the size of the body of the organisms. By contrast, the same granulometric curves for the pteropods shown in Fig. 1.54 are entirely different, climbing sharply until a strong maximum, reached for a size between 10 and 20. Together with other structural and shape-based features, these pattern spectra are used as input to a sophisticated classifier, which currently discriminates between 5 different types of organisms with over 90% accuracy. The ultimate goal of the project is to be able to classify more than a dozen different kinds of organisms with over 90% accuracy. For more details, refer to (Tang *et al* 1996).

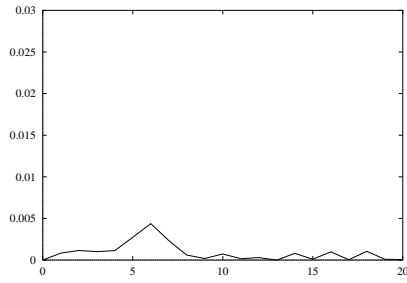
1.4.7 Final Notes on Granulometries

Even though the concept of granulometries was introduced close to three decades ago, the computation time required to extract granulometric curves has long made it impossible to use them for most practical applications. To address this issue, a comprehensive set of fast algorithms for computing granulometries in binary and grayscale images was proposed (Vincent 1994a, 1994b). These algorithms constituted the topic of the present section. Some of them are several orders of magnitudes faster than previously available techniques, so it now becomes possible to use granulometries on a “routine” basis. A number of examples of applications illustrated the use of these tools for the extraction of global size information

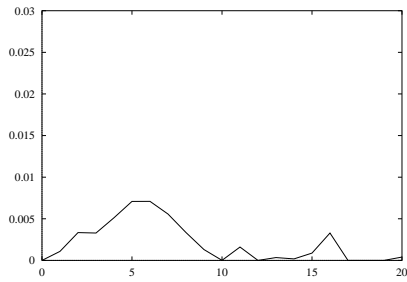
^{||} The support of the Office of Naval Research, through grant N00014-93-1-0606, is gratefully acknowledged for the plankton study.



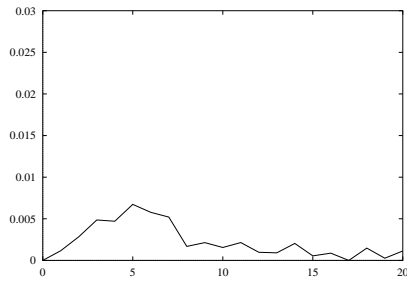
curve for (a)



curve for (b)

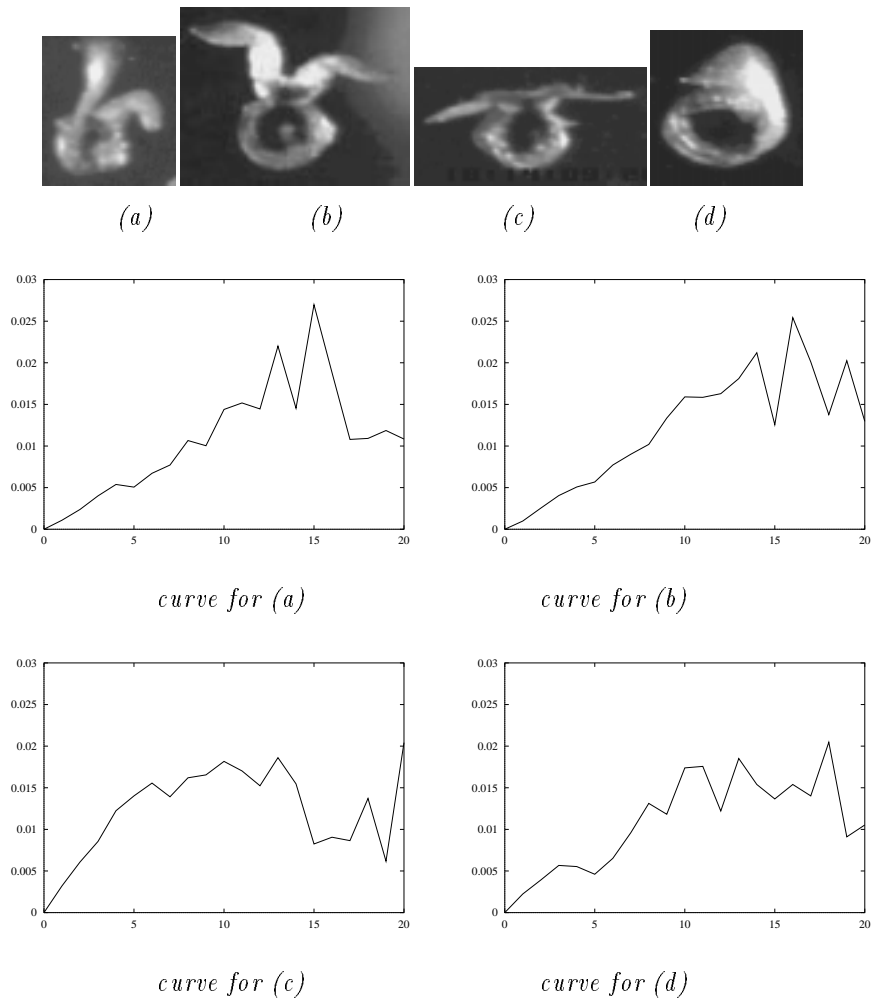


curve for (c)



curve for (d)

Figure 1.53 *Pattern spectra of copepod oithona*

Figure 1.54 *Pattern spectra of pteropods*

directly from grayscale or binary images, the extraction of granulometric feature vectors, and the characterization of textures.

One of the key new concepts that enabled the efficient computation of grayscale granulometries is that of *opening trees*: such structures are shown to provide a compact representation for the successive openings of a grayscale image by line segments of increasing size. They are at the heart of the algorithm proposed for grayscale granulometries with maxima of linear openings. In addition, while the efficient computation of *exact* grayscale granulometries with square openings remains an open problem, opening trees provide a way to extract pseudo-granulometries by minima of openings with line segments at different orientations, which can be used to approximate square granulometries. These techniques even extend to the fast computation of granulometries by area openings and closings (Vincent 1992b).

Beyond simply turning granulometries into a useful and computationally efficient set of tools, the algorithms described in the present section are expected to significantly contribute to expanding the range of problems that can be addressed using granulometries. For example, they can easily be extended to *local granulometries*, where they prove useful for texture segmentation and related problems, as described in (Vincent 1996). In addition, the concept of opening trees proves useful not only for granulometries, but for a variety of new operations such as grayscale *size transforms* (Vincent 1996). This new set of tools should therefore help popularize the use of grayscale granulometries and related techniques in the image and signal analysis community.

1.5 Conclusion

After some reminders on classic morphological image analysis concepts, operators, and notations, this chapter focused on two very important topics of morphology: segmentation and granulometries. Because of the introduction of new notions such as the dynamics and the opening trees, and because of advances in morphological algorithms (e.g., watersheds and granulometry algorithms), these two areas have boomed recently: an increasing number of papers is being published on watershed segmentation and derived techniques, and the same is true with granulometries. Moreover, the range of image analysis applications that are being approached using these techniques is growing steadily.

Two fundamental aspects of applied mathematical morphology were covered here: problem-solving (i.e., how to use the available tools to solve a particular image analysis problem), and algorithms (i.e., given a useful tool, how to implement it efficiently). The section on segmentation (Section 1.3) focused on the problem-solving aspect, whereas the section on granulometries (Section 1.4) focused on the algorithmic aspect. To further tie these two sections together, let us stress that the granulometry algorithms described enable the efficient and robust automatic extraction of size information directly from grayscale images. This information is often invaluable for segmentation: automatic extraction of structure size as part of a segmentation algorithm eliminates the need for “hard” size parameters or thresholds, thereby greatly enhancing robustness.

For more information on Mathematical Morphology in general, and in particular on the theoretical aspects of morphology, the reader is invited to refer to Serra (1982, 1988) and to Heijmans (1994). The reference section below also lists a number of papers and chapters that can be helpful to further probe any of the topics covered in this chapter. For a complete in-depth handbook of applied mathematical morphology, the reader may have to wait until 1997 or 1998 (Schmitt and Vincent 1997).

References

- G. Bertrand and X. Wang. An algorithm for a generalized distance transformation based on Minkowski operations. In *9th International Conference on Pattern recognition*, pages 1163–1167, Rome, Nov. 1988.
- S. Beucher. Watersheds of functions and picture segmentation. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pages 1928–1931, Paris, May 1982.
- S. Beucher and C. Lantuéjoul. Use of watersheds in contour detection. In *International Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation*, Rennes, France, 1979.
- S. Beucher and F. Meyer. The morphological approach to segmentation: the watershed transformation. In E. R. Dougherty, editor, *Mathematical Morphology in Image Processing*, pages 433–481. Marcel-Dekker, Sept. 1992.
- G. Birkhoff. *Lattice Theory*, volume 25. A.M.S. Colloq., third edition, 1983.
- G. Borgefors. Distance transformations in arbitrary dimensions. *Comp. Vis., Graphics and Image Processing*, 27:321–345, 1984.
- G. Borgefors. Distance transformations in digital images. *Comp. Vis., Graphics and Image Processing*, 34:334–371, 1986.
- L. Calabi and W. Harnett. Shape recognition, prairie fires, convex deficiencies and skeletons. Technical Report 1, Parke Math. Lab. Inc., One River Road, Carlisle MA, 1966.
- Y. Chen and E. Dougherty. Texture classification by gray-scale morphological granulometries. In *SPIE Vol. 1818, Visual Communications and Image Processing*, Boston MA, Nov. 1992.
- P. Danielsson. Euclidean distance mapping. *Comp. Graphics and Image Processing*, 14:227–248, 1980.
- C. S. Davis, S. M. Gallager, and A. R. Solow. Microaggregations of oceanic plankton observed by towed video microscopy. *Science*, 257:230–232, Jan. 1992.
- E. Dougherty, J. Pelz, F. Sand, and A. Lent. Morphological image segmentation by local granulometric size distributions. *Journal of Electronic Imaging*, 1(1), Jan. 1992.
- M. Grimaud. *La Géométrie Numérique en Morphologie Mathématique: Application à la Détection Automatique de Microcalcifications en*

- Mammographie Numérique*. PhD thesis, Ecole des Mines, Paris, Dec. 1991.
- M. Grimaud. A new measure of contrast: Dynamics. In *SPIE Vol. 1769, Image Algebra and Morphological Image Processing III*, pages 292–305, San Diego CA, July 1992.
- R. M. Haralick, S. Chen, and T. Kanungo. Recursive opening transform. In *IEEE Int. Computer Vision and Pattern Recog. Conference*, pages 560–565, Champaign IL, June 1992.
- R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*. Addison-Wesley, 1991.
- H. Heijmans. *Morphological Image Operators*. Academic Press, Boston, 1994.
- E. J. Kraus, H. Heijmans, and E. R. Dougherty. Gray-scale granulometries compatible with spatial scalings. *Signal Processing*, 34:1–17, 1993.
- C. Lantuéjoul. Issues of digital image processing. In R. M. Haralick and J.-C. Simon, editors, *Skeletonization in Quantitative Metallography*. Sijthoff and Noordhoff, Groningen, The Netherlands, 1980.
- C. Lantuéjoul and S. Beucher. On the use of the geodesic metric in image analysis. *Journal of Microscopy*, 121:39–49, Jan. 1981.
- C. Lantuéjoul and F. Maisonneuve. Geodesic methods in quantitative image analysis. *Pattern Recognition*, 17(2):177–187, 1984.
- B. Laÿ. Recursive algorithms in mathematical morphology. In *Acta Stereologica Vol. 6/III*, pages 691–696, Caen, France, Sept. 1987. 7th International Congress For Stereology.
- F. Maisonneuve. Extrema régionaux: Algorithme parallèle. Technical Report 781, Ecole des Mines, CGMM, Paris, 1982.
- P. Maragos. Pattern spectrum and multiscale shape representation. *IEEE Trans. Pattern Anal. Machine Intell.*, 11(7):701–716, July 1989.
- P. Maragos and R. Ziff. Threshold superposition in morphological image analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, 12(5), May 1990.
- G. Matheron. *Eléments pour une Théorie des Milieux Poreux*. Masson, Paris, 1967.
- G. Matheron. *Random Sets and Integral Geometry*. John Wiley and Sons, New York, 1975.
- F. Meyer. Contrast feature extraction. In J.-L. Chermant, editor, *Quantitative Analysis of Microstructures in Material Sciences, Biology and Medicine*, Stuttgart, FRG, 1978. Riederer Verlag. Special issue of Practical Metallography.
- H. Minkowski. Allgemein lehrrsätze über konvexe polyeder. *Nach. Ges. Wiss. Göttingen*, pages 198–219, 1897.
- L. Najman and M. Schmitt. A dynamic hierarchical segmentation algorithm. In J. Serra and P. Soille, editors, *EURASIP Workshop*

- ISMM'94, Mathematical Morphology and its Applications to Image Processing*, Fontainebleau, France, Sept. 1994. Kluwer Academic Publishers.
- J.-F. Rivest, P. Soille, and S. Beucher. Morphological gradients. In *SPIE/SPSE Vol. 1658, Nonlinear Image Processing III*, pages 139–150, Feb. 1992.
- A. Rosenfeld and J. Pfaltz. Sequential operations in digital picture processing. *J. Assoc. Comp. Mach.*, 13(4):471–494, 1966.
- A. Rosenfeld and J. Pfaltz. Distance functions on digital pictures. *Pattern Recognition*, 1:33–61, 1968.
- M. Schmitt. Variations on a theme in binary mathematical morphology. *Journal of Visual Communication and Image Representation*, 2(3):244–258, Sept. 1991.
- M. Schmitt and J. Mattioli. *Morphologie Mathématique*. Masson, Paris, 1994.
- M. Schmitt and L. Vincent. *Morphological Image Analysis: a Practical and Algorithmic Handbook*. Cambridge University Press, (to appear in 1997).
- J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.
- J. Serra, editor. *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances*. Academic Press, London, 1988.
- J. Serra and L. Vincent. An overview of morphological filtering. *Circuits, Systems and Signal Processing*, 11(1):47–108, Jan. 1992.
- S. R. Sternberg. Grayscale morphology. *Comp. Vis., Graphics and Image Processing*, 35:333–355, 1986.
- H. Talbot and L. Vincent. Euclidean skeletons and conditional bisectors. In *SPIE Vol. 1818, Visual Communications and Image Processing*, Boston, MA, Nov. 1992.
- X. Tang, K. Stewart, L. Vincent, H. Huang, M. Marra, S. Gallager, and C. Davis. Automatic plankton image recognition. *International Artificial Intelligence Review Journal*, 1996.
- C. Vachier and L. Vincent. Valuation of image extrema using alternating filters by reconstruction. In *SPIE Vol. 2568, Neural, Morphological, and Stochastic Methods in Image and Signal Processing*, pages 94–103, San Diego, CA, July 1995.
- L. Vincent. *Algorithmes Morphologiques à Base de Files d'Attente et de Lacets: Extension aux Graphes*. PhD thesis, Ecole des Mines, Paris, May 1990.
- L. Vincent. Efficient computation of various types of skeletons. In *SPIE Medical Imaging V*, San Jose, CA, 1991.
- L. Vincent. Exact euclidean distance function by chain propagations. In *IEEE Int. Computer Vision and Pattern Recog. Conference*, pages 520–525, Maui, HI, June 1991.

- L. Vincent. New trends in morphological algorithms. In *SPIE/SPSE Vol. 1451, Nonlinear Image Processing II*, pages 158–169, San Jose, CA, Feb. 1991.
- L. Vincent. Morphological area openings and closings for grayscale images. In *NATO Shape in Picture Workshop*, pages 197–208, Driebergen, The Netherlands, Sept. 1992.
- L. Vincent. Morphological grayscale reconstruction: Definition, efficient algorithms and applications in image analysis. In *IEEE Int. Computer Vision and Pattern Recog. Conference*, pages 633–635, Champaign IL, June 1992.
- L. Vincent. Grayscale area openings and closings: their efficient implementation and applications. In *EURASIP Workshop on Mathematical Morphology and its Applications to Signal Processing*, pages 22–27, Barcelona, May 1993.
- L. Vincent. Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE Transactions on Image Processing*, 2:176–201, Apr. 1993.
- L. Vincent. Fast grayscale granulometry algorithms. In J. Serra and P. Soille, editors, *EURASIP Workshop ISMM'94, Mathematical Morphology and its Applications to Image Processing*, pages 265–272, Fontainebleau, France, Sept. 1994. Kluwer Academic Publishers.
- L. Vincent. Fast opening functions and morphological granulometries. In *SPIE Vol. 2300, Image Algebra and Morphological Image Processing V*, pages 253–267, San Diego, CA, July 1994.
- L. Vincent. Local grayscale granulometries based on opening trees. In *ISMM'96, International Symposium on Mathematical Morphology*, Atlanta, GA, May 1996.
- L. Vincent and S. Beucher. The morphological approach to segmentation: an introduction. Technical report, Ecole des Mines, CMM, Paris, 1989.
- L. Vincent and E. R. Dougherty. Morphological segmentation for textures and particles. In E. R. Dougherty, editor, *Digital Image Processing Methods*, pages 43–102. Marcel-Dekker, New York, 1994.
- L. Vincent and B. Masters. Morphological image processing and network analysis of corneal endothelial cell images. In *SPIE Vol. 1769, Image Algebra and Morphological Image Processing III*, pages 212–226, San Diego, CA, July 1992.
- L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Machine Intell.*, 13(6):583–598, June 1991.
- P.-F. Yang and P. Maragos. Morphological systems for character image processing and recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages V.97–100, Minneapolis, MN, Apr. 1993.

- L.-P. Yuan. A fast algorithm for size analysis of irregular pore areas.
In *SPIE/SPSE Vol. 1451, Nonlinear Image Processing II*, pages 125–
136, San Jose, CA, Feb. 1991.