

Fast Granulometric Methods for the Extraction of Global Image Information

Luc Vincent

LizardTech, Inc.

4237 Manuela Avenue, Palo Alto CA 94306, USA

lvincent@lizardtech.com, luc@vincent-net.com

Abstract

Granulometries constitute one of the most useful and versatile sets of tools of morphological image analysis. They can be applied to a wide range of tasks, such as feature extraction, texture characterization, size estimation, image segmentation, etc., both for binary and for grayscale images. However, for most applications, traditional granulometry algorithms— involving sequences of openings or closings with structuring elements of increasing size—are prohibitively costly on non-specialized hardware. This has prevented granulometries from reaching a high level of popularity in the image analysis community.

This paper addresses the computational aspect of granulometries and proposes a comprehensive set of fast algorithms. In binary images, all but the simplest cases (namely linear granulometries based on openings with line segments) require the prior extraction of opening transforms (also referred to as “granulometry functions”). A very efficient algorithm is proposed for the computation of the most useful opening transforms. In grayscale images, linear granulometries are considered first and a particularly efficient algorithm is described. The concept of an *opening tree* is then proposed as a gray extension of the opening transform. It forms the basis of a novel technique for computing granulometries based on maxima of openings by line segments in different orientations, as well as *pseudo-granulometries* based on minima of linear openings.

The efficiency of this set of algorithms greatly increases the range of problems that can be addressed using granulometries. A number of application are used throughout the paper to illustrate the usefulness of the proposed techniques.

1 Introduction

The concept of a granulometry was introduced by George Matheron in the late sixties as a new tool for studying porous media [15] (See also [6] for related work). The size of the pores in such media was characterized using series of openings with structuring elements of increasing size [22]. The theoretical study of these operations led Matheron to propose the following definition:

Definition 1 Let $\Psi = (\psi_\lambda)_{\lambda \geq 0}$ be a family of image transformations depending on a parameter λ . This family constitutes a granulometry if and only if the following properties are satisfied:

$$\forall \lambda \geq 0, \quad \psi_\lambda \text{ is increasing,} \quad (1)$$

$$\forall \lambda \geq 0, \quad \psi_\lambda \text{ is anti-extensive,} \quad (2)$$

$$\forall \lambda \geq 0, \mu \geq 0, \quad \psi_\lambda \psi_\mu = \psi_\mu \psi_\lambda = \psi_{\max(\lambda, \mu)} \quad (3)$$

Property (3) implies that for every $\lambda \geq 0$, ψ_λ is an idempotent transformation, that is: $\psi_\lambda \psi_\lambda = \psi_\lambda$. The following characterization of granulometries can therefore be proposed:

Proposition 2 A family $\Psi = (\psi_\lambda)_{\lambda \geq 0}$ of image transformations depending on a unique parameter λ is a granulometry if and only if it forms a decreasing family of openings, that is:

$$\forall \lambda \geq 0, \quad \psi_\lambda \text{ is an opening,} \quad (4)$$

$$\forall \lambda \geq 0, \mu \geq 0, \quad \lambda \geq \mu \implies \psi_\lambda \leq \psi_\mu. \quad (5)$$

Recall that an opening is defined as an operation that is *idempotent*, *increasing*, and *anti-extensive*. For more information on openings, closings, and other morphological concepts, see [22, 23]. Another useful reference, focusing on the practical aspects of these operations, is [32].

The above proposition does not require the openings ψ_λ to be *morphological* openings, that is to be based on a structuring element [22]. *Algebraic* granulometries based on *algebraic* openings (See [23]) are also perfectly valid. Granulometries by area are an example of such algebraic granulometries. They are briefly discussed at the end of Section 4.

One of the most important results proved by Matheron in [16] is a characterization of granulometries based on morphological openings [22]:

Theorem 3 (Characterization) Let B be a compact set of \mathbb{R}^n . The family $\Gamma = (\gamma_\lambda)_{\lambda \geq 0}$ of openings by the homothetics $\lambda B = \{\lambda b \mid b \in B\}$, $\lambda \geq 0$, of B is a granulometry if and only if B is convex.

In practical terms, this theorem means that given a convex “primary grain” B , the family of openings with all the scales of B is a granulometry.

Before we focus this paper exclusively on the discrete case, two additional points need to be made:

- Everything that was said about granulometries so far applies to binary as well as grayscale images, in any dimension. From now on, for simplicity and readability of this paper, we focus on the 2-D case. However, the methods and algorithms described in the sequel extend to 3 or more dimensions, and can be restricted to 1-D signals. Both binary and grayscale images are considered in the paper.
- The type of granulometries considered until now are sometimes referred to as “granulometries by openings”. As often in morphology, dual operations can also be defined. We refer to these as “granulometries by closings” and use the following definition:

Definition 4 An increasing family $\Phi = (\phi_\lambda)_{\lambda \geq 0}$ of closings, that is such that:

$$\forall \lambda \geq 0, \mu \geq 0, \lambda \geq \mu \implies \phi_\lambda \geq \phi_\mu \quad (6)$$

is a granulometry by closings.

Recall that a closing is defined as an operation that is *idempotent*, *increasing*, and *extensive* [22, 23, 32]. In the rest of this paper, we mostly consider granulometries by openings. However, by duality, all the methods and algorithms described here extend to granulometries by closings: specifically, performing a granulometry by closings on an image is equivalent to performing a granulometry by openings on its complement.

Let us now specifically move to the digital case: from now on, all the transformations considered are acting on 2-D discrete images. A discrete image I is defined as a mapping from a bounded domain D_I , a (usually rectangular) subset of the discrete plane \mathbb{Z}^2 , into \mathbb{Z} . A binary image is such that its pixel values can only take value 1 (“ON” pixels) or 0 (“OFF” pixels). An equivalent representation for a binary image I is its set of ON pixels. Therefore, we often refer to binary images as sets of pixels in the discrete plane \mathbb{Z}^2 .

In this context, a granulometry is a sequence of openings γ_n , indexed on an *integer* $n \geq 0$. By definition, each opening is smaller than the previous one. In the binary case, this translates into:

$$\forall X \subset \mathbb{Z}^2, \forall n \geq m \geq 0, \gamma_n(X) \subseteq \gamma_m(X). \quad (7)$$

On the other hand, in the grayscale case, it means that:

$$\forall I, \forall n \geq m \geq 0, \gamma_n(I) \leq \gamma_m(I), \quad (8)$$

Where $\gamma_n(I) \leq \gamma_m(I)$ means that for any pixel p in domain D_I , $\gamma_n(I)(p) \leq \gamma_m(I)(p)$.

The granulometric analysis of a set X (i.e., a 2-D binary image) with respect to a granulometric family of openings $(\gamma_n)_{n \geq 0}$ is often compared to a *sieving* process: X is sieved

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 sieved is characteristic of this set and provides a “signature” of X with respect to the granulometry used.

In order to quantify the rate at which X is being sieved, a measure needs to be used for sets. Denote this measure by $m(A)$, for a given set A . In the vast majority of cases, $m(A)$ is taken to be the *area* of A , i.e., its number of pixels. In rare cases, other measures, such as the number of connected components in A , can be of interest. For details, see [4]. Given m , we can now propose the following definition:

Definition 5 The *granulometric curve or pattern spectrum* [14] of a set X with respect to a granulometry $\Gamma = (\gamma_n)_{n \geq 0}$ is the mapping $PS_\Gamma(X)$ given by:

$$\forall n > 0, PS_\Gamma(X)(n) = m(\gamma_n(X)) - m(\gamma_{n-1}(X)). \quad (9)$$

By duality, the above definition extends to granulometries by closings. Furthermore, pattern spectra can also be defined for grayscale images: in this context, the measure $m(I)$ of a grayscale image I is usually taken to be the sum of its pixel values, sometimes referred to as the “volume” of I . From now on, we exclusively use the area or number of pixels as a measure for binary images, and the volume or sum of pixel values as a measure for grayscale images.

In practice, the most useful granulometries are based on openings (or closings) with the homothetics of a simple convex structuring element B . Typically, B is a line segment, a square, or a hexagon. In such cases, we talk about “linear granulometries”, “square granulometries”, or “hexagonal granulometries” respectively. Granulometries based on maxima of openings with line segments at different orientations are also commonly used.

Fig. 1 shows an example of granulometric analysis. The input image, located in the top-left corner, is a binarized image of coffee beans. Some of its openings with squares of increasing size are shown in Fig. 1a, and the corresponding pattern spectrum is shown in Fig. 1b. This pattern spectrum exhibits a very well marked peak at opening size 20, corresponding to a 21×21 square. This can be interpreted as the *dominant* size of the beans (as size of the largest square a bean can contain) in this image. Granulometries therefore allow us to extract size information without any need for prior segmentation: the beans in this image exhibit significant overlap, yet their size can be estimated without individually identifying each bean. As we show later in the paper, this direct size estimation can also be performed on gray images directly, with a high level of efficiency.

Granulometries have been used for a variety of other image analysis tasks, including shape characterization and feature extraction [21, 34, 24], texture classification [3, 25, 8], and even segmentation [7]. However, until recently, granulometric analysis involved performing a series of openings and/or closings of increasing size, which is prohibitively expensive for most

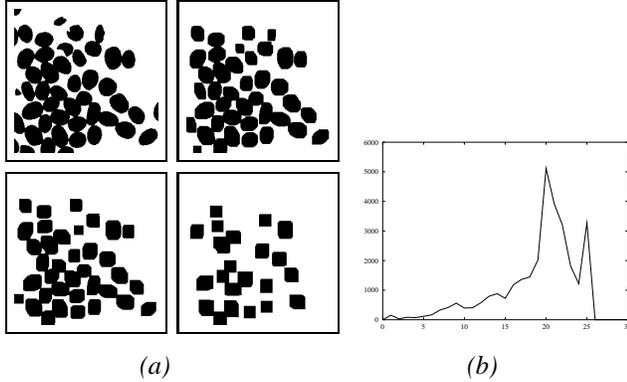


Figure 1: (a) Successive openings of Fig. 2a using squares of increasing size as structuring elements. (b) Corresponding granulometric curve, or pattern spectrum: the peak at size 20 is indicative of the typical size of the beans in this image.

applications, unless dedicated hardware is used. This is what motivated the work reported in the present paper, where we describe a number of new efficient granulometry algorithms.

In the next section, we first briefly review the literature on granulometry algorithms. A comprehensive set of fast binary granulometry algorithms is then proposed: section 2.2 is concerned with the simple case of linear granulometries. We show that their computation comes down to run-length extraction. In section 2.3, more complex cases, such as granulometries with squares, are discussed. The corresponding algorithms involve the determination of *granulometry functions*, for which fast new methods are proposed. Following the result in proposition 7’s result, pattern spectra are then derived by simple histogramming.

The grayscale case is considered next: in section 3, an extremely efficient method is proposed for computing linear grayscale granulometries. The algorithm is orders of magnitude faster than previously available techniques. It makes it therefore possible to use granulometries where previously unthinkable. We illustrate this point by using this new algorithm to efficiently extract size information directly from grayscale images. The concept of an *opening tree* is then presented in section 4: it provides a way to compactly encode all successive linear openings of a grayscale image. It forms the central idea of a new set of algorithms for computing granulometries with maxima of linear openings, as well as approximations of square granulometries. The methods are again several orders of magnitude faster than traditional techniques.

2 Binary Granulometry Algorithms

2.1 Background, Granulometry Functions

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 paper. But no matter how efficient an opening algorithm is, computing a pattern spectrum using a sequence of openings of increasing size 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 $\gamma_n(X)$ (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.

By comparison, very few algorithms have been proposed for the efficient computation of granulometries and pattern spectra. Before the grayscale granulometry algorithms described in the present paper were introduced in 1994 [30], all known methods dealt with the binary case exclusively. These methods all use *granulometry functions* as an intermediate step, so we now briefly review this concept. Given a set X (binary image) and a granulometry $\Gamma = (\gamma_n)_{n \geq 0}$ indexed on an integer parameter n , by definition $(\gamma_n(X))_{n \geq 0}$ is a decreasing sequence of sets:

$$\forall n \geq m \geq 0, \quad \gamma_n(X) \subseteq \gamma_m(X).$$

It is therefore possible to condense the representation of this sequence of sets by mapping each pixel p of X to the size n of the first opening that “removes” p . This defines a mapping from X to \mathbb{Z}^+ called *granulometry function*, and often referred to as *opening function* or *opening transform*). More formally, the following definition can be proposed:

Definition 6 (Granulometry Function) *Let I be a discrete binary image defined on domain D_I , and let $\Gamma = (\gamma_n)_{n \geq 0}$ be a granulometry indexed on an integer parameter n . The corresponding granulometry function $G_\Gamma(I)$ is defined by:*

$$\forall p \in D_I, \quad G_\Gamma(I)(p) = \min\{n \geq 0 \mid \gamma_n(I)(p) = 0\}. \quad (10)$$

Obviously, for any $n \geq 0$, the threshold of $G_\Gamma(X)$ above a value n is equal to $\gamma_n(X)$, the n -th opening of X :

$$\gamma_n(X) = \{p \in X \mid G_\Gamma(X)(p) > n\}. \quad (11)$$

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

Proposition 7 *The pattern spectrum $PS_\Gamma(X)$ of X for granulometry $\Gamma = (\gamma_n)_{n \geq 0}$ can be derived from the granulometry function $G_\Gamma(X)$ as follows:*

$$\forall n > 0, \quad PS_\Gamma(X)(n) = \text{card}\{p \mid G_\Gamma(X)(p) = n\}, \quad (12)$$

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

An example of a square granulometry function is shown in Fig. 2. The histogram of this granulometry function directly provides the pattern spectrum previously shown in Fig. 1b.

Granulometry functions would be of moderate interest if it weren’t for the fact that efficient methods exist to compute

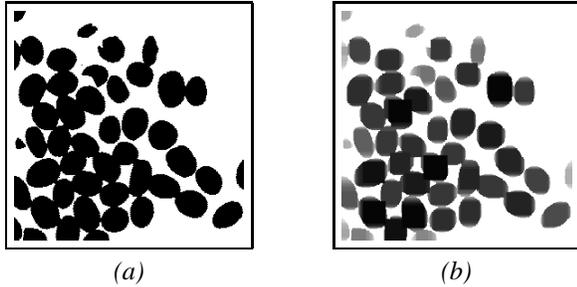


Figure 2: (a) original binary image of coffee beans; (b) square granulometry function of this image, in which dark regions correspond to higher pixel value.

them. Let us now briefly review published granulometry function algorithms: the algorithm proposed by Yuan [35] 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. An example is shown in Fig. 3 (See [26] 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 . While faster than a “brute force” approach to granulometries, this algorithm still requires a significant amount of image scans. In addition, the more complicated the image is or the larger the objects in it, the longer this method takes.

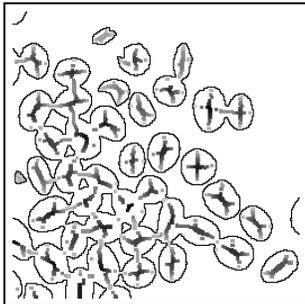


Figure 3: Quench function of Fig. 2a, dilated for clarity (dark pixels correspond to large values of the quench function).

Suprisingly, a better algorithm can be found in an earlier paper by Laÿ [12], in which the author devotes a few lines to the description of a sequential algorithm [17, 27] based on the distance function [18, 2], 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. The methods described in section 2.3 are very similar to this algorithm and extend it to other types of binary granulometries and granulometry functions.

The algorithm proposed in 1992 by Haralick *et al* [9] is attractive in that it allows in principle to compute granulometry functions with respect to any family of homothetic elements

(the base element or primary grain does not even have to be convex). 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 relatively slow.

We now describe a new comprehensive set of fast binary granulometry algorithms. In their simplest form (i.e., when using line segments as structuring elements for the underlying family of openings), there is no need to use granulometry functions, and the best methods simply rely on extraction and histogramming of run-lengths in specific orientations (see Section 2.2). Section 2.3 starts by covering the case of granulometries based on maxima of linear openings, which requires the computation of linear granulometry functions as an intermediate step. The more complex 2-D case is discussed next and granulometry function algorithms are described that offer a significant speed advantage over other published methods.

2.2 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:

$$L_n = \underbrace{\bullet \bullet \bullet \dots \bullet \bullet}_{n+1 \text{ pixels}} \quad (13)$$

From now on, we use the convention that the center of a structuring element is marked using a thicker dot than is used for 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 $N_0(p)$, $N_1(p)$, ..., $N_7(p)$, and the eight elementary directions are encoded in the following way:

$$\begin{array}{ccc} 3 & 2 & 1 \\ 4 & \bullet & 0 \\ 5 & 6 & 7 \end{array} \quad (14)$$

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 :

$$N_d^{(0)}(p) = p, \quad \text{and} \quad k > 0 \implies N_d^{(k)}(p) = N_d(N_d^{(k-1)}(p)). \quad (15)$$

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

Definition 8 The ray in direction d at pixel p in set X is given by:

$$r_{X,d}(p) = \{N_d^{(k)}(p) \mid k \geq 0 \text{ and } \forall 0 \leq j \leq k, N_d^{(j)}(p) \in X\}. \quad (16)$$

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 \bar{d} .

Definition 9 The run in direction d at pixel p in set X is given by:

$$R_{X,d}(p) = r_{X,d}(p) \cup r_{X,\bar{d}}(p). \quad (17)$$

The number of pixels in a run R will be called *length* of this run and denoted by $l(R)$.

The following proposition is immediate:

Proposition 10 The opening of X by L_n , denoted $X \circ L_n$, is the union of the horizontal runs $R_{X,0}(p)$ whose length is strictly greater than n :

$$X \circ L_n = \bigcup_{p \in X} \{R_{X,0}(p) \mid l(R_{X,0}(p)) > n\}. \quad (18)$$

Therefore, any horizontal run of length n is left unchanged by all 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:

$$\forall n > 0, \quad \text{PS}_0(X)(n) = \text{card}\{p \in X \mid l(R_{X,0}(p)) = n\}. \quad (19)$$

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

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. 4a, which is a binary image of lamellar eutectics. In [20], M. Schmitt 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 knowledge of the 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. 4b, and its peak at 3 indicates that the typical width of the lamellae is of 3 pixels.

2.3 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 $(\gamma_n)_{n \geq 0}$ where for each n , γ_n

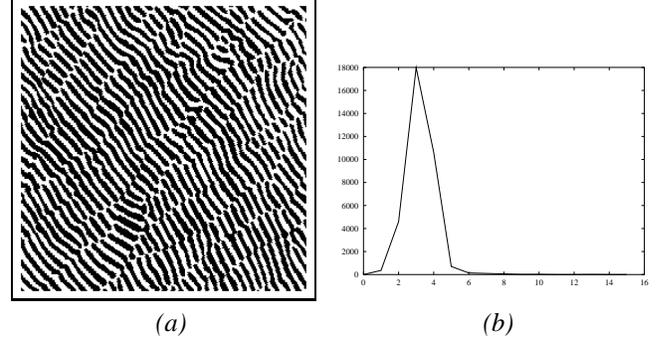


Figure 4: (a) Binary image of lamellar eutectics. (b) Its granulometric curve using line segments at +45 degrees orientation. This curve clearly indicates that the dominant width of the lamellae is of 3 pixels.

is an algebraic opening defined as the maximum of opening with horizontal segment L_n and opening with vertical segment L_n^\perp (where L_n^\perp denoted the vertical equivalent of L_n). 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 $(\gamma_n)_{n \geq 0}$ of the 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 11 Let $\Gamma^1 = (\gamma_n^1)_{n \geq 0}$ and $\Gamma^2 = (\gamma_n^2)_{n \geq 0}$ be two granulometries. Then, $\max(\Gamma^1, \Gamma^2) = (\max(\gamma_n^1, \gamma_n^2))_{n \geq 0}$ is also a granulometry and for any set X :

$$G_{\max(\Gamma^1, \Gamma^2)}(X) = \max(G_{\Gamma^1}(X), G_{\Gamma^2}(X)). \quad (20)$$

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 also gets tagged with its length $l(R)$. This involves scanning the black pixels of the image twice, and the white pixels only once. The resulting algorithm is hardly more time consuming than the one described in the previous section.

The case of truly 2-D binary granulometry functions is the next level in complexity. In the rest of this section, we first focus on granulometry functions $G_S(X)$ based on openings with the homothetics of the elementary square S ; then, we deal with the case of granulometry functions $G_D(X)$ based on the elementary “diamond” shape D :

$$S = \begin{array}{cc} \bullet & \bullet \\ \bullet & \bullet \end{array} ; \quad D = \begin{array}{ccc} & \bullet & \\ \bullet & \bullet & \bullet \\ & \bullet & \end{array} \quad (21)$$

(Note that the center of structuring element D is located on the bottom pixel of D instead of being at the geometric center.) Together with the linear case, these granulometries cover the vast majority of practical needs.

Like Haralick's algorithm [9], the first step of the present method consists of computing what some authors have called a *generalized distance function* [1, 10]. Let B be an arbitrary structuring element containing its center. Let \oplus denote the traditional Minkowski addition operator [22] defined by

$$X \oplus Y = \{x + y \mid x \in X, y \in Y\}, \quad (22)$$

and 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 [22]:

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

Definition 12 *The generalized distance function $d_B(X)$ with respect to the family of structuring elements $(nB)_{n \geq 0}$ assigns to each pixel $p \in X$ the smallest $k > 0$ such that $p \notin \varepsilon_{kB}(X)$:*

$$d_B(X)(p) = \min\{k > 0 \mid p \notin \varepsilon_{kB}(X)\}. \quad (23)$$

Generalized distance functions are determined using sequential algorithms that are straightforwardly derived from the original algorithm proposed by Rosenfeld [17, 18]. When the center of the structuring element is at 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. For example, in the case when $B = S$ (see Eq. (21)), the following algorithm can be proposed:

Algorithm: Generalized distance function 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 a generalized distance function resulting from this algorithm is shown in Figs. 6a–b. A way to interpret the result is to say that, for each pixel p , if one was to translate structuring element $(d_S(X)(p) - 1)S$ so that its center coincides with p , this translated element—denoted $p + (d_S(X)(p) - 1)S$ —would be entirely included in X . However, $p + d_S(X)(p)S$ would not be contained in X . We can therefore state the following proposition:

Proposition 13 *The granulometry function $G_S(X)$ is obtained from the distance function $d_S(X)$ as follows:*

$$\forall p \in X, \quad G_S(X)(p) = \max\{d_S(X)(q) \mid p \in (q + (d_S(X)(q) - 1)S)\}. \quad (24)$$

In algorithmic terms, we can compute $G_S(X)$ by propagating the value $d_S(X)(p)$ of each pixel p over the square $p + (d_S(X)(p) - 1)S$, and then by taking the pixelwise maximum of the values propagated at each pixel. This is illustrated by Fig. 5.

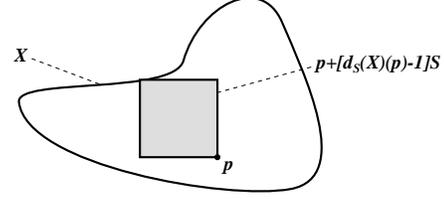


Figure 5: Given a set X and a pixel $p \in X$, $p + (d_S(X)(p) - 1)S$ is the largest square included in X , with p being the bottom-right pixel. For any pixel q in this square, the value taken by the granulometry function $G_S(X)$ at q is greater or equal to $d_S(X)(p)$.

In the technique proposed by Haralick *et al* [9], 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 computationally expensive, and in the case of the 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 (See Eq. (22) of the two elementary line segments E_1 and E_2 :

$$S = \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \oplus \begin{array}{c} \bullet \\ \bullet \end{array} = \begin{array}{c} \bullet \\ \bullet \end{array} \oplus \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} = E_1 \oplus E_2 \quad (25)$$

It follows that the complex propagation step of the granulometry function algorithm described in [9] can in fact be decomposed into two much simpler propagations, with substantial gain in speed: 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 new value v . The algorithm maintains an array `propag` containing the number of times each value remains to be propagated.

Algorithm: Left propagation of distance values of $d_S(X)$

- Input: image I of the generalized distance function $d_S(X)$;
- For each line of the image, do:
 - Initializations: `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`[$I(p)$] $\leftarrow I(p)$;

```

For  $i = 0$  to  $\text{maxval}$ , do:
     $\text{propag}[i] \leftarrow \text{propag}[i] - 1$ ;
     $\text{maxval} \leftarrow \text{largest } i \leq \text{maxval} \text{ such}$ 
    that  $\text{propag}[i] \geq 0$ ;
     $I(p) \leftarrow \text{maxval}$ 

```

A few implementation tricks can speed up computation by substantially reducing the number of times the entire array `propag` is scanned for each line. While their description would be beyond the scope of this paper, note that in the optimized algorithm, the `propag` array needs to be scanned only in rare cases. For most pixels, only a comparison and an assignment are performed. Therefore, 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). Using again the coffee bean image as running example, the result of this propagation step is shown in Figs. 6c–d, and the final granulometry function obtained after upward propagation in each column is shown in Figs. 6e–f.

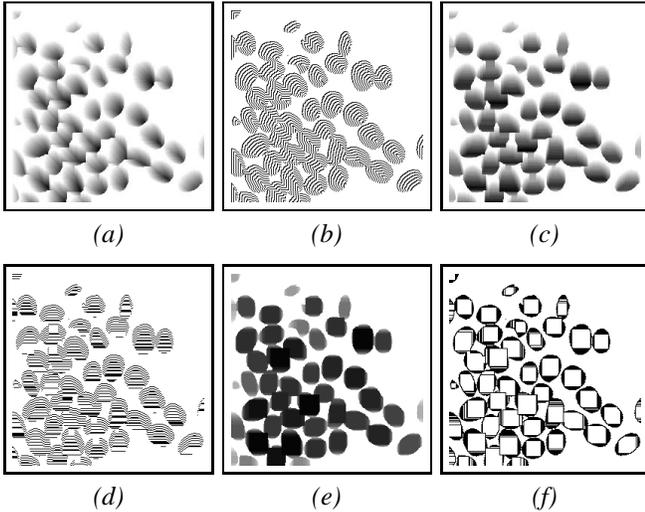


Figure 6: Computation of granulometry function using square structuring elements. (a) generalized distance function; (b) level lines; (c) propagation of values from right to left; (d) level lines; (e) final granulometry function; (f) level lines of granulometry function.

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. (25) and (26)). Furthermore, it extends to the computation of hexagonal granulometry functions in the hexagonal grid [12].

In addition, in square grids, the generalized distance function $d_D(X)$ with respect to “diamond” structuring element D is easily computed using an algorithm very similar to that presented for structuring element S . Unfortunately, D cannot be decomposed as the Minkowski addition of elementary line segments. The closest “approximation” is obtained with

$$D = \begin{array}{c} \bullet \\ \bullet \bullet \bullet \\ \bullet \end{array} \neq \begin{array}{c} \bullet \oplus \bullet \\ \bullet \oplus \bullet \\ \bullet \oplus \bullet \end{array} = E_3 \oplus E_4 \quad (26)$$

and does not contain the central pixel of D ! Therefore, starting from distance function $d_D(X)$ and using the propagation algorithm in the Southwest-Northeast (SW-NE) direction, then in the Southeast-Northwest (SE-NW) direction, results in an incorrect propagation function, as illustrated in Fig. 7. Correct “diamond” granulometry functions can nevertheless be obtained with this technique if SW-NE and SE-NW propagation steps are followed by a “hole-filling” step in which each pixel p such that

$$\forall i \in \{0, 2, 4, 6\}, \quad I(N_i(p)) = I(p) + 1$$

is given value $I(p)+1$. An example of such granulometry function is shown in Fig. 8.

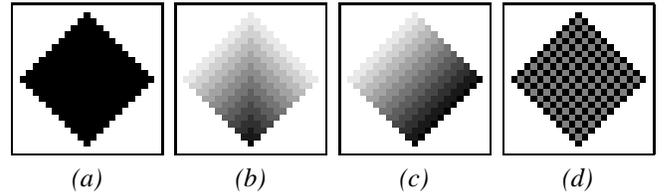


Figure 7: When using “diamonds” as structuring elements, the two propagation steps of the granulometry function algorithm (steps (b) and (c) above) need to be followed by a “hole-filling” step (d).

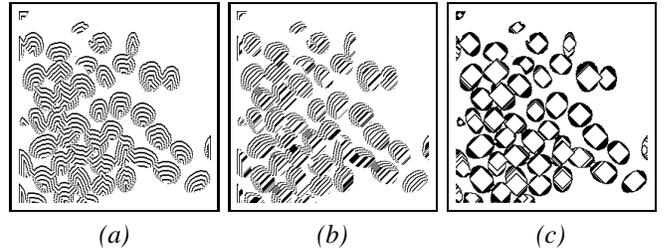


Figure 8: Computation of granulometry function using “diamond” structuring elements. (a) level lines of generalized distance function; (b) level lines of relief obtained after SW to NE propagation; (c) level lines of final granulometry function

2.4 Summary on binary granulometry algorithms

Table 1 summarizes the speed of these granulometry functions on the 256×256 coffee bean image used as a 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 type of implementation. Note however that for this coffee bean image, which has approximately 30,000 black pixels, Haralick’s algorithm [9] takes between 0.5s and 0.6s to compute the square granulometry function shown in Fig. 6e, 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 in the present paper is between three and four times faster.

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

Table 1: Execution time of various granulometry function algorithms on the 256×256 coffee bean image, measured on a Sun Sparc Station 10 workstation.

3 Linear Grayscale Granulometries

Grayscale granulometries are potentially even more useful than binary ones, because they enable extraction of information directly from grayscale images. A number of theoretical results have been published on them (see e.g. [11]); however, until the algorithms described in the present paper were first published [30], no efficient technique was available to compute grayscale granulometries. These tools have therefore not been used very much in practice. The methods proposed in this section and in the next one aim to remedy this situation. The focus of the present section is on linear granulometries, so the algorithms described are directly applicable to 1-D signals as well.

Like in Section 2.2, let us assume for simplicity that the line segments used as structuring elements are horizontal (the algorithm easily extends to any orientation). The structuring elements considered are the L_n 's of equation (13).

Let us now analyze the effect of an opening by L_n , $n \geq 0$, on a grayscale image I . Denote by $N_l(p)$ and $N_r(p)$ respectively the left and the right neighbors of a pixel p , and consider the following definitions:

Definition 14 A horizontal line segment S , of length $l(S) = n$, is a set of pixels $\{p_0, p_1, \dots, p_{n-1}\}$ such that for $0 < i < n$, $p_i = N_r(p_{i-1})$.

Definition 15 A horizontal maximum M of length $l(M) = n$ in a grayscale image I is a horizontal line segment $\{p_0, p_1, \dots, p_{n-1}\}$ such that:

$$\forall i, 0 < i < n, \quad I(p_i) = I(p_0) \quad \text{and} \\ I(N_l(p_0)) < I(p_0), \quad I(N_r(p_{n-1})) < I(p_0). \quad (27)$$

This notion is the 1-D equivalent of the classic *regional maximum* concept [13]. The study of how such maxima are altered through horizontal openings is at the basis of the algorithm described in the present section. Denote by $I \circ B$ the standard morphological opening of an image I by a structuring element B . The following proposition holds:

Proposition 16 Let $M = \{p_0, p_1, \dots, p_{n-1}\}$ be a horizontal maximum of I , of length $l(M) = n$. Let $p \in M$.

$$\forall k < n, \quad (I \circ L_k)(p) = I(p), \quad (28) \\ \text{for } k = n, \quad (I \circ L_n)(p) = \max\{I(N_l(p_0)), I(N_r(p_{n-1}))\} \\ \forall k > n, \quad (I \circ L_k)(p) < I(p). \quad (30)$$

That is, 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 $\max\{I(N_l(p_0)), I(N_r(p_{n-1}))\}$. In granulometric terms, it means that the contribution of maximum M to the n -th bin of the horizontal pattern spectrum $\text{PS}_h(I)$ is:

$$n \times [I(p) - \max\{I(N_l(p_0)), I(N_r(p_{n-1}))\}]. \quad (31)$$

This is illustrated in Fig. 9.

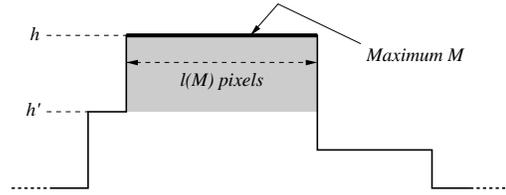


Figure 9: Horizontal cross section of I with a maximum M . The shaded area, of volume $(h - h') \times l(M)$ shows the local contribution of this maximum to the $l(M)$ -th bin of the horizontal pattern spectrum.

In addition, the local effect of the horizontal opening of size n on maximum M results in a new “plateau” P of pixels being created at altitude $\max\{I(N_l(p_0)), I(N_r(p_{n-1}))\}$. This plateau contains M and may or may not 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 as follows: horizontal lines of I are considered one after the other and scanned from left to right. Each horizontal maximum M of the current line is identified, and its contribution to $\text{PS}_0(I)(l(M))$ 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 in Fig. 10.

Specifically, the algorithm works as follows, on each scanline:

Algorithm: horizontal granulometry for a line of image I

- Initialize pattern spectrum: for each $n > 0$, $\text{PS}[n] \leftarrow 0$
- For each maximum M of this line (in any order) do:
 - Let $P \leftarrow M$ be the current maximum considered
 - While P is a horizontal maximum, do:

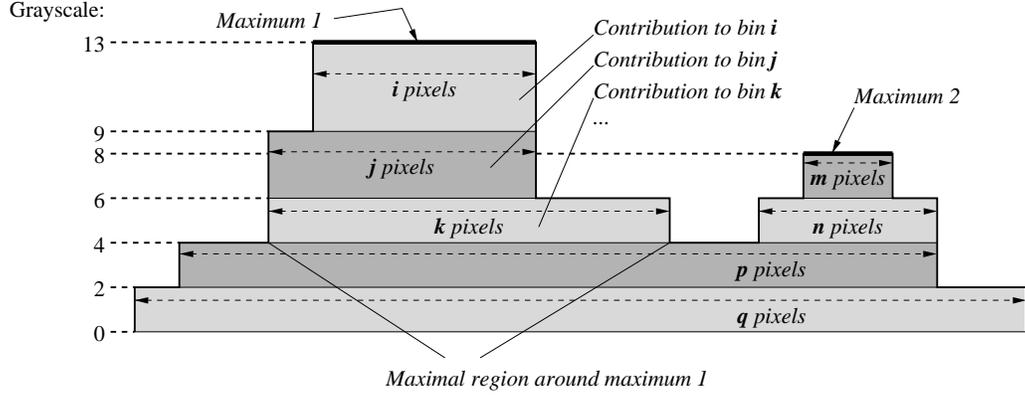


Figure 10: Illustration of a horizontal grayscale granulometry algorithm for a line with two maxima. In this case, maxima were scanned from left to right. Processing maximum 1 results in bins i , j , and k of the pattern spectrum being incremented. While processing maximum 2, bins m and n are first incremented, 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.

```

 $p_l \leftarrow N_l(P)$ , neighbor of  $P$  to the
left;
 $p_r \leftarrow N_r(P)$ , neighbor of  $P$  to the
right;
 $n \leftarrow l(P)$ , length of horizontal
maximum  $P$ ;
Add contribution of maximum  $P$  to
 $n$ -th bin of pattern spectrum:
 $PS[n] \leftarrow$ 
 $PS[n] + n \times (I(P) - \max(I(p_l), I(p_r)))$ ;
For any pixel  $p$  in  $P$ :
 $I(p) \leftarrow \max(I(p_l), I(p_r))$ ;
 $P \leftarrow$  new plateau of pixels formed
after opening of size  $n$  of  $P$ ;
- Put special marker on left and right
of current plateau  $P$  so that while
processing subsequent maxima, we
already know that this region is a
plateau and can skip over it;
```

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. 10). 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 2, the new algorithm described in this section is orders of magnitude faster.

The speed of this new algorithm opens up a range of new applications for grayscale granulometries. Traditionally, the practical problems addressed by granulometries had to do with ei-

| | Traditional | Proposed algorithm |
|---------|-------------|--------------------|
| Image a | 204s | 0.248s |
| Image b | 204s | 0.206s |

Table 2: Execution time of traditional opening-based technique and of proposed algorithm for the computation of a grayscale granulometry with horizontal line segments, computed for size 1 to 512. The 512×512 weld images of Fig. 11 were used for this comparison, done on a Sun Sparc Station 10.

ther texture discrimination or with 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 the algorithm described in this section, 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 by 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 the algorithm.

Figs. 11a–b are used to illustrate how grayscale granulometries can be used to estimate size information¹. 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. 11a–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. 11c–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 Images gracefully provided by CMIS, CSIRO, Australia.

11a and 11b do not have any preferential orientation. Second, the curves in Fig. 11c exhibit a well-marked peak for size 4, whereas the peak of the curves in Fig. 11d is found for size 12. We conclude that the typical width and height of the patterns in Fig. 11a and in Fig. 11b is of 4 pixels and 12 pixels respectively. Additionally, Fig. 11d shows that the distribution of pattern sizes in Fig. 11b is a lot wider than in Fig. 11a.

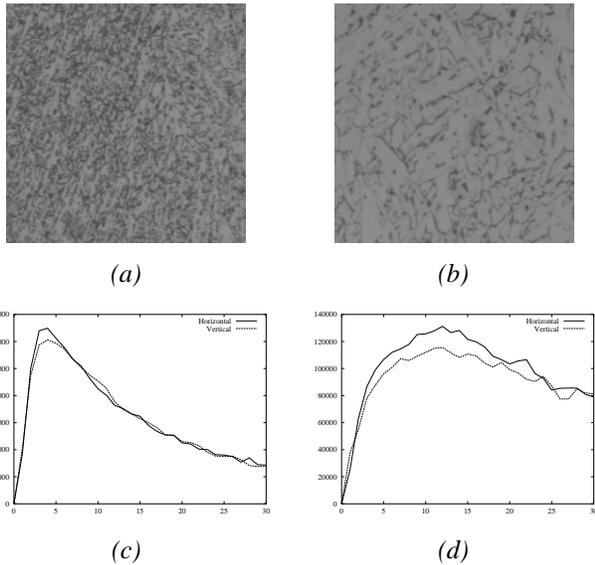


Figure 11: Using linear grayscale granulometries to estimate object size without prior segmentation. Curve (c) clearly indicates that the dominant width and height of the white patterns in image (a) is 4 pixels. Similarly, curve (d) shows that the dominant width/height of the patterns in (b) is 12.

Another example is shown in Fig. 12. Images 12a and 12b show X-rays of a healthy lung and of a lung exhibiting signs of *silicosis*, respectively. *Silicosis* is a medical condition also known as “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 12b different from image 12a. 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 for discriminating between these two types of textures. To that end, M. Grimaud, author of the original (unpublished) study, had used grayscale granulometries by openings and closings with squares. The resulting pattern spectra for both X-ray images, shown in Figs. 12a–b, are extremely different from one another. However, in order to unambiguously discriminate between these two images, expensive grayscale granulometries with squares are not necessary: using the algorithm presented here to extract *horizontal* grayscale granulometric curves is just as effective, as shown in Fig. 12d, and is orders of magnitude faster. Note that this would not necessarily be true in more complex cases, and 2-D granulometries, or even granulometries using grayscale structuring elements, remain essential.

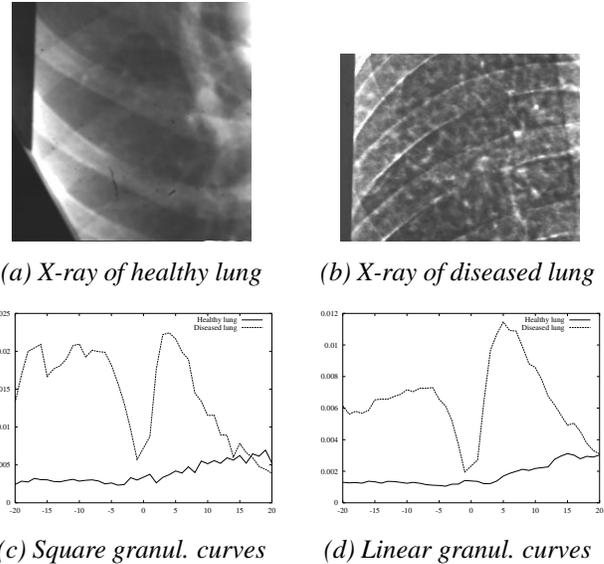


Figure 12: Example of texture differentiation using grayscale granulometries by openings and closings. The pattern spectra of image (a) and image (b) are completely different, whether linear or square granulometries are used. The curves in (c) and (d) are such that positive indices correspond to granulometries by openings whereas negative indices correspond to granulometries by closings.

4 Opening Trees and Grayscale Granulometries

In order to deal with more complicated cases, we now introduce a technique that can be seen as a grayscale generalization of the concept of granulometry functions. When performing openings of increasing size of a binary image, each “on” pixel p is turned “off” for an opening size given by the value of the granulometry function at pixel p . In other words, the granulometry function encodes for each pixel the successive values it takes for increasing opening sizes, specifically in this case, a series of 1’s followed by a series of 0’s). Similarly for grayscale images, as the size of the opening increases, the value of each pixel decreases monotonically. That is, for a granulometry $\Gamma = (\gamma_n)_{n \geq 0}$, a gray image I and a pixel p in I :

$$I(p) = \gamma_0(I)(p) \geq \gamma_1(I)(p) \geq \gamma_2(I)(p) \geq \dots$$

Given this sequence of numbers, the granulometric contribution of p can be derived straightforwardly. So if this sequence of numbers was available for any pixel p in I , the corresponding pattern spectrum would be easily derived.

Unfortunately, even if it were possible to compute such lists of values quickly, assigning one to each image pixel would require a huge amount of memory. A more compact representation therefore needs to be designed, that takes into account the intrinsic “redundancy” of opened images, characterized by their large plateaus of pixels. Such a compact representation cannot easily be designed for any kind of opening. However, if we again consider linear openings and the associated linear

granulometries, an elegant solution can be proposed to both the problem of computing these lists of values, and the problem of storing them compactly. By extension, as mentioned at the end of the present section, the technique proposed below can be applied to granulometries based on *connected openings and closings* [19].

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

$$(I \circ L_n)(p) < I(p) \iff \exists \text{ horiz maximum } M, l(M) = n \text{ and } p \in M. \quad (32)$$

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 at size n . For larger opening sizes, this list may itself be merged with other lists, etc.

Another way to explain this is as follows: given any two pixels p and q on line L of image I , if we repeatedly open I with line segments of increasing length, at some point, p and q will become part of the same “plateau” of pixels. As soon as this happens, p and q will always have the same value for any subsequent opening in the series, so the lists of values taken by these two pixels through openings of monotonically increasing size can be merged past this point. This is illustrated by Fig. 13.

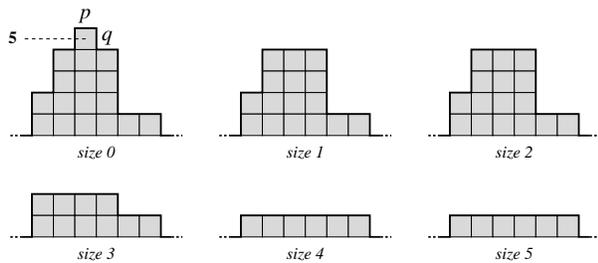


Figure 13: Linear openings of increasing size (0 through 5) on an image line represented by its cross-section. As soon as pixels p and q become part of the same plateau of pixels, which here happens for opening size 1, the values taken by p and q for subsequent opening sizes are identical.

In addition, in all practical cases, the series $((I \circ L_n)(p))_{n \geq 0}$ of values taken by p through successive openings exhibits large plateaus of constant value. In other words, for any given opening size n , there is a range $[n_1, n_2]$ of sizes, $n_1 \leq n \leq n_2$ such that:

$$\forall m, n_1 \geq m \geq n_2, (I \circ L_m)(p) = (I \circ L_n)(p).$$

In practice, such plateaus tend to get wider and wider with increasing opening sizes. So to represent $((I \circ L_n)(p))_{n \geq 0}$ more compactly, we can simply keep the list of opening sizes n such that $(I \circ L_n)(p) < (I \circ L_{n-1})(p)$, together with the associated opening value $(I \circ L_n)(p)$.

We now exploit the two previous remarks through the novel concept of an *opening tree* [30], which is central to the algorithms described later in this section. With each image line, a tree T is associated such that:

- The leaves of T are the image pixels.
- 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. Reaching a pair (h_i, n_i) means that for opening size n_i , the value of p is equal to h_i . By convention, for any pixel p , $(h_0, n_0) = (I(p), 0)$. In addition, the following relationship holds:

$$\forall i > 0, h_i > h_{i+1} \text{ and } n_i < n_{i+1}. \quad (33)$$

Now, one can easily verify that for $n \geq 0$, the value of the opening of size n of I at pixel p is given by:

$$(I \circ L_n)(p) = h_j, \text{ where } j \text{ is such that } n_j \leq n < n_{j+1}. \quad (34)$$

Such trees efficiently exploit the redundancies described earlier. They can be extracted using an algorithm very similar to the one described in the previous section: the only difference is that instead of dynamically updating a granulometric array, we now build opening trees. An example of an opening tree is shown in Fig. 14.

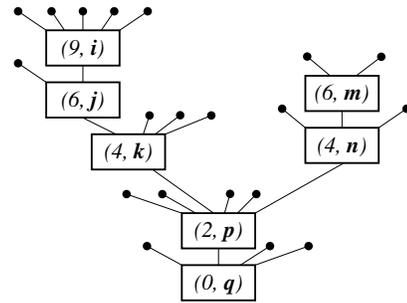


Figure 14: Opening tree representation of the image line of Fig. 10. The leaves of this tree (marked by \bullet) correspond to the image pixels.

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 granulometry function (See Section 2.3). One can prove that, in the worst case, the opening tree has one node per image pixel. In practice though, only 0.3 to 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 trees (one tree per scanline). In addition, the horizontal pattern spectrum of I , $PS_h(I)$, can be computed from T as follows:

Algorithm: horizontal granulometry of I from its opening tree representation

- 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 grayscale granulometries than the one proposed in Section 3. 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 of a grayscale image I , the following approach should be used:

1. Compute the set of “horizontal opening trees” of I , that is, extract an opening tree description for each image line. After this step, each pixel of I becomes the leaf of a horizontal opening tree, and the successive values taken by this pixel through horizontal openings of increasing size are obtained by following the tree down to its root.
2. Compute the set of “vertical opening trees” of I , thereby describing each image column as an opening tree.
3. For each pixel p of I , follow the associated horizontal opening tree T_1 and vertical opening tree T_2 down to their respective roots, computing the desired values along the way and updating the pattern spectrum $PS(I)$.

Specifically, this last step is accomplished by using the following algorithm:

Algorithm: granulometry of I from opening 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:

```

      (h1, n1) ← next node down in T1;
      while n2 ≤ size and (h2, n2) exists
      do:
        (h2, n2) ← next node down in T2;
        PS(I)(size) ←
        PS(I)(size) + (v - max(h1, h2));
        v ← max(h1, h2);
  
```

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 in Table 3. In this table, a granulometry by maxima of linear openings in four 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.

| | Traditional | New algorithm |
|---------|-------------|---------------|
| Image a | 13mn 44s | 2.973s |
| Image b | 13mn 24s | 2.780s |

Table 3: Execution time of traditional opening-based technique compared to present algorithm for the computation of a granulometry by maxima of linear openings in 4 orientations. Like in Table 2, the 512×512 weld images of Fig. 11 were used for this comparison, done on a Sun Sparc Station 10. Granulometries were computed for opening sizes 1 to 512.

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 [16], minima of openings with line segments of increasing length do constitute a decreasing family of image operators. The resulting pseudo-granulometric curves often characterize similar image features as square granulometries would.

The extraction of these pseudo-granulometries can be done exactly as outlined earlier, except that step #3 in the algorithm (extraction of granulometric values from opening trees) needs to be adjusted. The following algorithm should be used. It can also be extended to work with any number of opening trees:

Algorithm: pseudo-granulometry of I from opening 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 ;
 - $level_1 \leftarrow h_1$;
 - $level_2 \leftarrow h_2$;

```

- while  $(h_1, n_1)$  OR  $(h_2, n_2)$  exist, do:
  size  $\leftarrow \min(n_1, n_2)$ ; ( $n_1$  if  $(h_2, n_2)$ 
  does not exist, and vice versa);
  if  $(h_1, n_1)$  exists and size =  $n_1$  do:
    level1  $\leftarrow h_1$ 
     $(h_1, n_1) \leftarrow$  next node down in  $T_h$ ;
  else:
    level1  $\leftarrow +\infty$ 
  if  $(h_2, n_2)$  exists and size =  $n_2$  do:
    level2  $\leftarrow h_2$ 
     $(h_2, n_2) \leftarrow$  next node down in  $T_v$ ;
  else:
    level2  $\leftarrow +\infty$ 
  PS( $I$ )(size)  $\leftarrow$ 
  PS( $I$ )(size) +  $(v - \min(\text{level}_1, \text{level}_2))$ ;
   $v \leftarrow \min(\text{level}_1, \text{level}_2)$ ;

```

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 images² [5, 24]. As an illustration, Fig. 15 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. 16 are entirely different, climbing sharply to 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 [24].

Let us conclude this section by generalizing opening trees (resp. closing trees) to other forms of openings (resp. closings). The reason we were able to condense the information contained in the successive linear openings of an image line into an opening tree is that, once two pixels become part of the same plateau (also known as “flat zone”), they are always part of the same plateau for any subsequent opening size. Therefore, beyond linear openings, any granulometry based on *two-dimensional* openings that preserve flat zones lends itself to an opening tree representation. These openings are sometimes called *connected openings* [19].

More formally, given an image I , call *plateau* of I any connected set of pixels P (usually 4-connected or 8-connected) such that for any two pixels x and y in P , $I(x) = I(y)$. The following can be stated:

Proposition 18 *Let $\Gamma = (\gamma_n)_{n \geq 0}$ be a granulometry based on connected openings, that is, such that for any $n \geq 0$, for any image I :*

P connected plateau of I

²The support of the Office of Naval Research, through grant N00014-93-1-0606, is gratefully acknowledged for the plankton study.

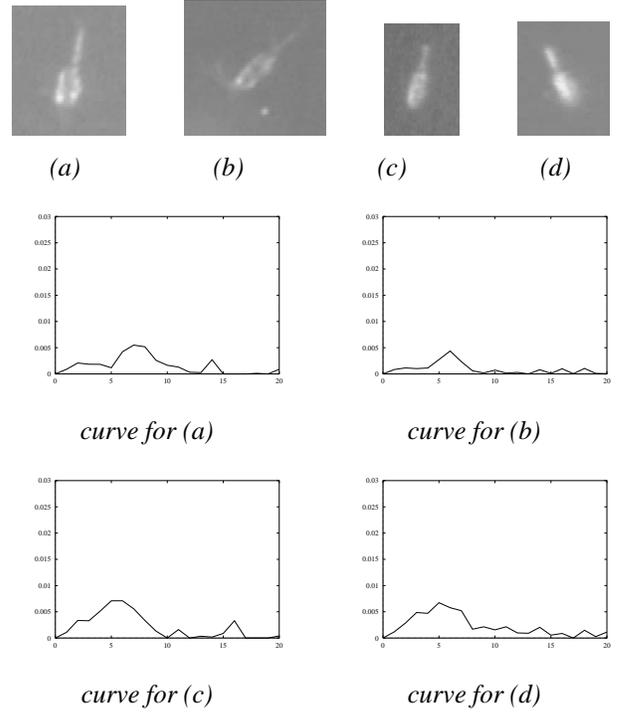


Figure 15: Pattern spectra of copepod oithona

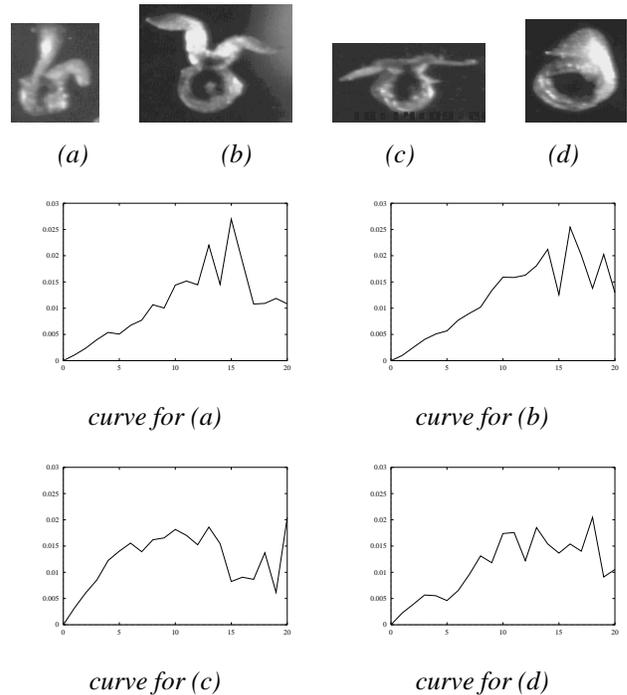


Figure 16: Pattern spectra of pteropods

$$\implies P \text{ connected plateau of } \gamma_n(I). \quad (35)$$

Then the decreasing image sequence $(\gamma_n(I))_{n \geq 0}$ can be represented as an opening tree.

Note that this result extends to N -dimensional images. In particular, it means that the methods and algorithms presented in the present section, as well as in Section 3, extend to these connected granulometries. *Granulometries by area*, that is, granulometries based on area openings or closings [28] are a specific example of connected granulometries. The area opening algorithm proposed in [29] can be easily combined with the methods described in the present paper to enable computation of area granulometries. However, the practical use of these operations remains unclear.

5 Conclusion and Future Work

Although the concept of granulometry was introduced over three decades ago, the computation time required to extract granulometric curves made it impossible to use granulometries for most practical applications, even using today's powerful computers. The main goal of the present paper was to remedy this situation by proposing a comprehensive set of efficient granulometry algorithms. The improvement provided by some of these new methods went beyond original expectations, with speed gains of three orders of magnitude or more in some cases.

The efficiency of the algorithms described here make it now possible to use morphological granulometries on a routine basis. Furthermore, these techniques open new areas of application for these tools. For example, they can be used to directly extract *global* estimates of object size in grayscale images without need for any segmentation or binarization. Extracting such information can be a goal in itself, but it can also be essential to calibrate the parameters of subsequent image analysis algorithms to be applied, thereby enhancing overall system robustness. Numerous examples of applications were shown in the paper, illustrating the usefulness of granulometries for such tasks as size estimation, feature extraction, and texture characterization and segmentation. These new algorithms, and their useful new applications, should contribute to popularize the use of granulometries for a growing number of image and signal analysis problems.

One of the key concepts proposed here is that of an *opening tree*: such structures were shown to provide a compact representation for the successive openings of a grayscale image by line segments of increasing size. They are at the core of the algorithm proposed for grayscale granulometries with maxima of linear openings. In addition, 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. Finally, opening trees and derived algorithms can be extended to the fast computation of "connected" granulometries, and to novel local granulometric transforms, such as adaptive openings and size transforms, as described in [31, 33].

Undoubtedly, the spectrum of powerful image representation schemes provided by opening trees remains to be fully exploited. Our expectation is that opening trees could be used on the one hand to speed up computation of a number of standard morphological transforms, and on the other hand to derive new useful image operators. A very interesting and challenging open problem remains: is it possible to generalize opening trees into "opening graphs" or other structures that would enable compact representation of any granulometric sequence of openings for grayscale images? Hopefully, a solution to this problem will be proposed soon, thereby extending the reach of present methods to the full spectrum of commonly used granulometries, in particular grayscale square granulometries.

References

- [1] 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.
- [2] G. Borgefors. Distance transformations in digital images. *Comp. Vis., Graphics and Image Processing*, 34:334–371, 1986.
- [3] Y. Chen and E. Dougherty. Texture classification by gray-scale morphological granulometries. In *SPIE Vol. 1818, Visual Communications and Image Processing*, pages 931–942, Boston MA, Nov. 1992.
- [4] M. Coster and J.-L. Chermant. *Précis d'Analyse d'Images*. CNRS Ed., Paris, 1985.
- [5] 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.
- [6] Definer. A generalization of the concept of size. *Journal of Microscopy*, 95:203–216, 1971.
- [7] E. Dougherty, J. Pelz, F. Sand, and A. Lent. Morphological image segmentation by local granulometric size distributions. *Journal of Electronic Imaging*, 1(1):46–60, Jan. 1992.
- [8] C. Gratin, J. Vitrià, F. Moreso, and D. Serón. Texture classification using neural networks and local granulometries. In J. Serra and P. Soille, editors, *EURASIP Workshop ISMM'94, Mathematical Morphology and its Applications to Image Processing*, pages 309–316, Fontainebleau, France, Sept. 1994. Kluwer Academic Publishers.
- [9] 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.
- [10] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*. Addison-Wesley, 1991.

- [11] E. J. Kraus, H. Heijmans, and E. R. Dougherty. Grayscale granulometries compatible with spatial scalings. *Signal Processing*, 34:1–17, 1993.
- [12] 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.
- [13] F. Maisonneuve. Extrema régionaux: Algorithme parallèle. Technical Report 781, Ecole des Mines, CGMM, Paris, 1982.
- [14] P. Maragos. Pattern spectrum and multiscale shape representation. *IEEE Trans. Pattern Anal. Machine Intell.*, 11(7):701–716, July 1989.
- [15] G. Matheron. *Éléments pour une Théorie des Milieux Poreux*. Masson, Paris, 1967.
- [16] G. Matheron. *Random Sets and Integral Geometry*. John Wiley and Sons, New York, 1975.
- [17] A. Rosenfeld and J. Pfaltz. Sequential operations in digital picture processing. *J. Assoc. Comp. Mach.*, 13(4):471–494, 1966.
- [18] A. Rosenfeld and J. Pfaltz. Distance functions on digital pictures. *Pattern Recognition*, 1:33–61, 1968.
- [19] P. Salembier and J. Serra. Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Transactions on Image Processing*, 4(8):1153–1160, 1995.
- [20] M. Schmitt. Variations on a theme in binary mathematical morphology. *Journal of Visual Communication and Image Representation*, 2(3):244–258, Sept. 1991.
- [21] M. Schmitt and J. Mattioli. Shape recognition combining mathematical morphology and neural networks. In *SPIE: Application of Artificial Neural Network*, Orlando, FL, Apr. 1991.
- [22] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.
- [23] J. Serra, editor. *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances*. Academic Press, London, 1988.
- [24] X. Tang, K. Stewart, L. Vincent, H. Huang, M. Marra, S. Gallager, and C. Davis. Automatic plankton image recognition. *Artificial Intelligence Review*, 12:177–199, 1998.
- [25] M. Vanrell and J. M. Vitria. Mathematical morphology, granulometries, and texture perception. In *SPIE Vol. 2030, Image Algebra and Morphological Image Processing IV*, pages 152–161, San Diego CA, June 1993.
- [26] L. Vincent. Efficient computation of various types of skeletons. In *SPIE Vol. 1445, Medical Imaging V*, pages 297–311, San Jose, CA, 1991.
- [27] L. Vincent. New trends in morphological algorithms. In *SPIE/SPSE Vol. 1451, Nonlinear Image Processing II*, pages 158–169, San Jose, CA, Feb. 1991.
- [28] L. Vincent. Morphological area openings and closings for grayscale images. In *NATO Shape in Picture Workshop*, pages 197–208, Driebergen, The Netherlands, Sept. 1992. Springer Verlag.
- [29] 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.
- [30] 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.
- [31] L. Vincent. Local grayscale granulometries based on opening trees. In P. Maragos, R. Schafer, and M. Butt, editors, *ISMM'96, International Symposium on Mathematical Morphology*, pages 273–280, Atlanta, GA, May 1996. Kluwer Academic Publishers.
- [32] L. Vincent. Current topics in applied morphological image analysis. In W. Kendall, O. Barndorff-Nielsen, and M. van Lieshout, editors, *Current Trends in Stochastic Geometry and its Applications*. Chapman & Hall, London, 1997.
- [33] L. Vincent. Granulometries and opening trees. *Fundamenta Informaticae*, 41(1–2):57–90, Jan. 2000.
- [34] P.-F. Yang and P. Maragos. Morphological systems for character image processing and recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume V, pages 97–100, Minneapolis, MN, Apr. 1993.
- [35] 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.