[Proc. SPIE Vol. 2300 "Image Algebra and Morphological Image Processing V", pp. 253–267, San Diego CA, July 1994]

Fast Opening Functions and Morphological Granulometries

Luc Vincent
Xerox Imaging Systems
9 Centennial Drive, Peabody, MA 01960

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, from feature extraction, to texture characterization, to size estimation, to image segmentation, etc. However, traditional granulometry algorithms—involving sequences of openings or closings with structuring elements of increasing size—are prohibitively costly on non-specialized hardware. This problem has prevented granulometries from reaching a high level of popularity in the image analysis community.

In this paper, a comprehensive set of fast algorithms for computing granulometries in binary images is first proposed: linear granulometries (i.e., granulometries based on openings with line segments) constitute the easiest case, and are computed using image "run-length". The 2-D case (granulometries with square or "diamond"-shaped structuring elements, or granulometries with unions of line-segments at different orientations) involves the determination of opening functions or granulometry functions. The grayscale case is then addressed, and a new algorithm for computing grayscale linear granulometries is introduced. This algorithm is orders of magnitude faster than any previously available technique. The techniques introduced in this paper open up a new range of applications for granulometries, examples of which are described in the paper.

1 Introduction

The concept of granulometry was introduced by G. Matheron in 1967 [10] 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 [15]. The theoretical study of these operations led Matheron to propose the following definition:

Definition 1 *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:*

$$\forall \lambda \geq 0, \qquad \phi_\lambda \text{ is increasing,} \tag{1}$$

$$\forall \lambda \geq 0, \qquad \phi_\lambda \text{ is anti-extensive,} \tag{2}$$

$$\forall \lambda \geq 0, \mu \geq 0, \qquad \phi_\lambda \phi_\mu = \phi_\mu \phi_\lambda = \phi_{\max(\lambda,\mu)} \tag{3}$$

Property (3) 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 [15]. 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 [11].

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:

$$\forall X, \quad \forall n \geq m \geq 0, \quad \phi_n(X) \subseteq \phi_m(X). \tag{4}$$

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 2 *The granulometric curve or pattern spectrum [9] of a set X with respect to a granulometry $\Phi = (\phi_n)_{n\geq 0}$ is the mapping $PS_\Phi(X)$ given by:*

$$\forall n > 0, \quad PS_\Phi(X)(n) = m(\phi_n(X)) - m(\phi_{n-1}(X)). \tag{5}$$

Since $(\phi_n(X))_{n\geq 0}$ is a decreasing sequence of sets $(\phi_0(X) \supseteq \phi_1(X) \subseteq \phi_2(X) \supseteq \cdots)$, it is possible to condense its representation by introducing the concept of granulometry function [8, 19, 5]:

Definition 3 *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 $x \notin \phi_n(X)$:*

$$x \in X \longmapsto G_\Phi(X)(x) = \min\{n > 0 \mid x \notin \phi_n(X)\}. \tag{6}$$

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

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

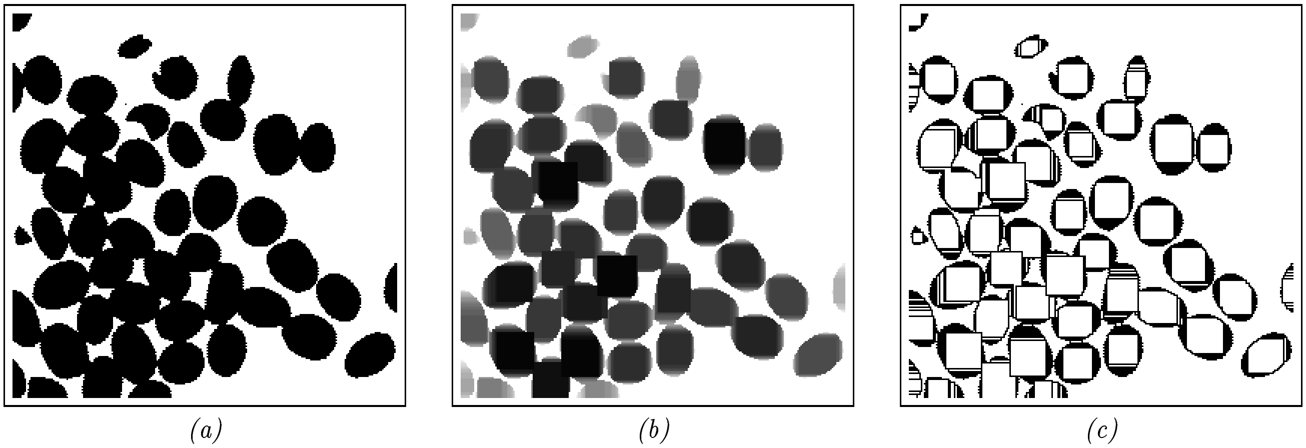
The following property follows immediately and states that the pattern spectrum can be obtained as histogram of the granulometry function:

Proposition 4 *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:*

$$\forall n > 0, \quad PS_\Phi(X)(n) = \mathrm{card}\{p \mid G_\Phi(X)(p) = n\}, \tag{7}$$

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

The concept of granulometry function is central to the algorithms described in section 4. An example of square granulometry function is shown in Fig. 1.



(a) *(b)* *(c)*

Figure 1 : (a) original binary image of coffee beans, (b) square granulometry function of this image, in which dark regions correspond to higher pixel value (c) level lines of the granulometry function.

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 [19]; 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. 1a with respect to a family of openings with squares (as was used for Fig. 1) is shown in Fig. 2. From the observed pattern spectrum, the typical size of the beans (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 overlapping, yet their size can be estimated without individually identifying each bean.

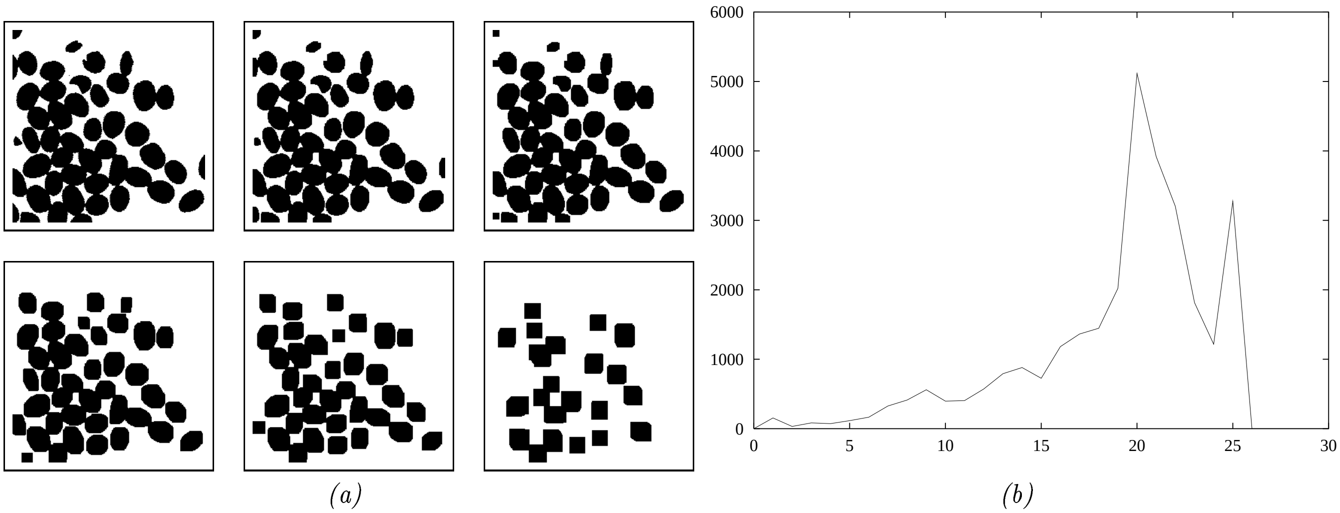


Figure 2 : *(a) Successive openings of a Fig. 1 using squares of increasing size as structuring elements. (b) Corresponding granulometric curve, or pattern spectrum: the peak at size 20 is indicative of the typical size of the beans in original image.*

Granulometries have been used for a variety of other image analysis tasks, including shape characterization and feature extraction (see for example [20]), texture classification (see [3]), and even segmentation (see for example [4]). 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 harware is used.

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 3 is concerned with the simple case of linear granulometries. In section 4, more complex cases, such as granulometries with squares, are discussed. The corresponding algorithms involve the determination of granulometry functions, for which fast algorithms are proposed. Following prop. 4's result, pattern spectra are then derived by simple histogramming,

Lastly, the grayscale case is considered in section 5 a new algorithm is introduced for computing linear grayscale granulometries. The algorithm is several orders of magnitude faster than any previouly available technique. 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 a grayscale image.

2 Background, Previous Work

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 used, determining a patterm 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 [21] for determining binary square granulometries consists in 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 [16] 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.

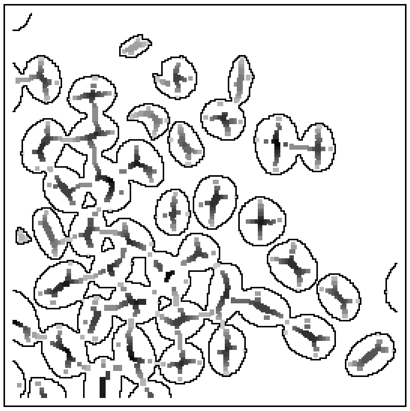


Figure 3 : *Quench function of Fig. 1a, dilated for clarity; dark skeleton pixels correspond to large values of the quench function.*

Suprisingly, a better algorithm can be found in an earlier paper by Laÿ [8], in which the author devotes a few lines to the description of a sequential algorithm [12, 17] based on the distance function [13, 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. In section 4, this technique is for the first time described in detail, and is extended to other types of binary granulometries.

The algorithm proposed in 1992 by Haralick *et al* is interesting in that it allows in principle to compute granulometry functions with respect to any family of homothetic elements[1]. 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.

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:

$$L_n \quad = \quad \underbrace{\bullet \;\bullet \;\bullet \cdots \bullet \;\large\bullet}_{n+1 \text{ pixels}} \tag{8}$$

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

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 $N_0(p)$, $N_1(p)$,..., $N_7(p)$, and the eight elementary directions are encoded in the following way:

$$\begin{array}{ccc} 3 & 2 & 1 \\ 4 & \bullet & 0 \\ 5 & 6 & 7 \end{array} \tag{9}$$

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:

$$N_d^{(0)}(p) = p, \quad \text{and} \quad k > 0 \Longrightarrow N_d^{(k)}(p) = N_d(N_d^{(k-1)}(p)). \tag{10}$$

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

Definition 5 *The* ray *in direction d at pixel p in set X is given by:*

$$r_{X,d}(p) = \{N_d^{(k)}(p) \mid k \geq 0 \text{ and } \forall 0 \geq j \geq k, N_d^{(j)}(p) \in X\}. \tag{11}$$

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

$$R_{X,d}(p) = r_{X,d}(p) \cup r_{X,\check{d}}(p). \tag{12}$$

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

The following proposition is immediate:

Proposition 7 *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:*

$$X \circ L_n = \bigcup_{p \in X} \{R_{X,0}(p) \mid l(R_{X,0}(p)) > n\}. \tag{13}$$

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:

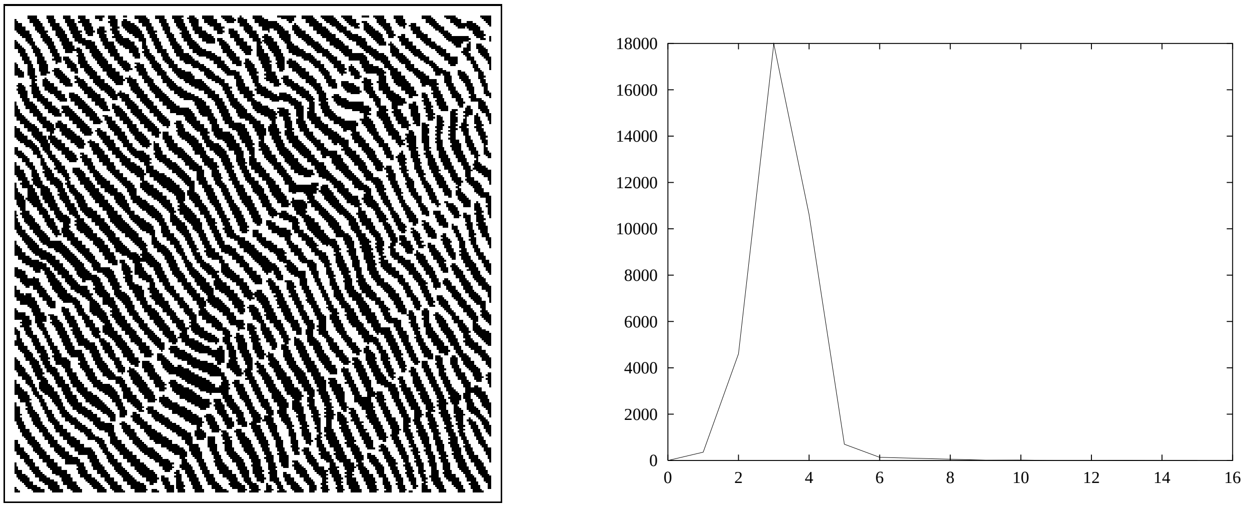
$$\forall n > 0, \quad \mathrm{PS}_0(X)(n) = \mathrm{card}\{p \in X \mid l(R_{X,0}(p)) = n\}. \tag{14}$$

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

Algorithm: horizontal binary granulometry

```
• Initialize pattern spectrum:  for each $n > 0$, PS[$n$] $\leftarrow$ 0
• Scan each line of image from left to right.
• In this process, each time a run $R$ is discovered, do:
        PS[$l(R)$] $\leftarrow$ 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 [14], 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 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. 4b, and its peak at 3 indicates that the typical width of the lamellae is of 3 pixels.



(a)

(b)

Figure 4 : *Binary image of lamellar eutectics (a) and its granulometric curve using line segments at +45 degrees orientation (b)*

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 8 *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:*

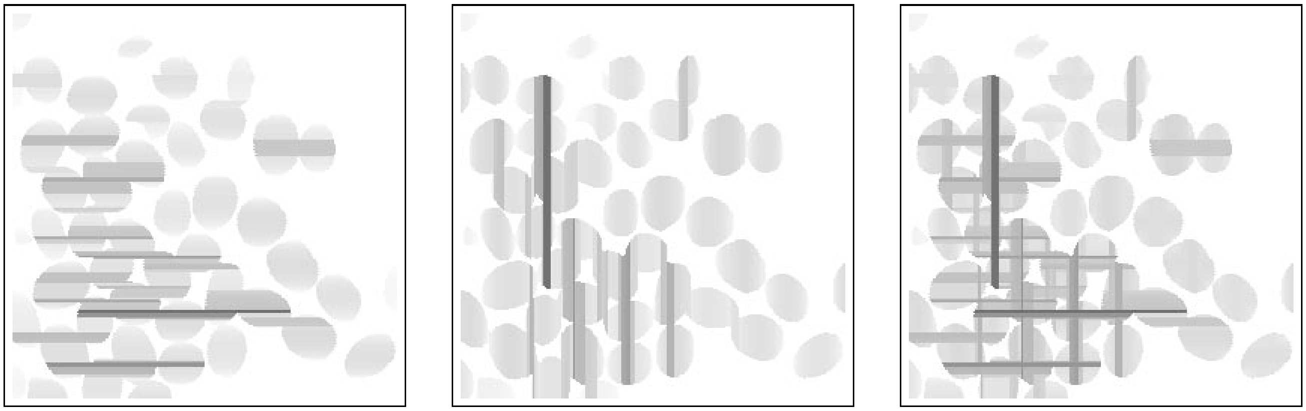
$$G_{\max(\Phi,\Psi)}(X) = \max(G_\phi(X), G_\psi(X)). \tag{15}$$

Determining the linear granulometry function of a binary image is a relatively straightforward task. Take for example the horizontal case: like in 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 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. Examples of linear granulometry functions are shown in Fig. 5.

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

$$S \;=\; \begin{array}{cc} \bullet & \bullet \\ \bullet & {\Large\bullet} \end{array} \quad ; \quad D \;=\; \begin{array}{ccc} & \bullet & \\ \bullet & \bullet & \bullet \\ & {\Large\bullet} & \end{array} \tag{16}$$

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



(a) (b) (c)

Figure 5 : *(a) Horizontal granulometry function; (b) vertical granulometry functions. (c) Pointwise maximum of these two images provides the granulometry function corresponding to maxima of openings with vertical and horizontal line segments.*

Like Haralick's algorithm [5], the first step of the present one consists in computing what some authors have called a *generalized distance function* [1, 6]. Let B be an arbitrary structuring element containing its center. Let

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

denote the structuring element "of size n". Let also ε_B denote the erosion by structuring element B [15]:

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

Definition 9 *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 $p \notin \varepsilon_{kB}(X)$:*

$$d_B(X)(p) = \min\{k > 0 \mid p \notin \varepsilon_{kB}(X). \tag{17}$$

Generalized distance functions are determined using sequential algorithms that are straightforwardly derived from the original algorithm proposed by Rosenfeld [12, 13]. 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. (16)), 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 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 $[d_S(X)(p)]S$ 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 10 *The granulometry function $G_S(X)$ is obtained from $d_S(X)$ as follows:*

$$\forall p \in X, \quad G_S(X)(p) = \max\{d_S(X)(q) \mid p \in (q + d_S(X)(q)S)\}. \tag{18}$$

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

In the technique proposed by Haralick *et al* [5], 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:

$$S \;=\; \begin{array}{cc}\bullet & \bullet\\ \bullet & \Large\bullet\end{array} \;=\; \begin{array}{cc}\bullet & \Large\bullet\end{array} \;\oplus\; \begin{array}{c}\bullet\\ \Large\bullet\end{array} \;=\; E_1 \;\oplus\; E_2 \tag{19}$$

Therefore, the complex propagation step of the granulometry function algorithm described in [5] 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 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 $I(p) >$ maxval:
                maxval $\leftarrow I(p)$;
            propag[$I(p)$] $\leftarrow I(p)$;
            $\forall 0 < i <=$ maxval,  propag[$i$] $\leftarrow$ propag[$i$] $- 1$;
            maxval $\leftarrow$ largest $i \leq$ maxval such that propag[$i$] $>= 0$;
            $I(p) \leftarrow$ 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 paper. 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.

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. (19) and (20)). Furthermore, it extends to the computation of hexagonal opening functions in the hexagonal grid [8].

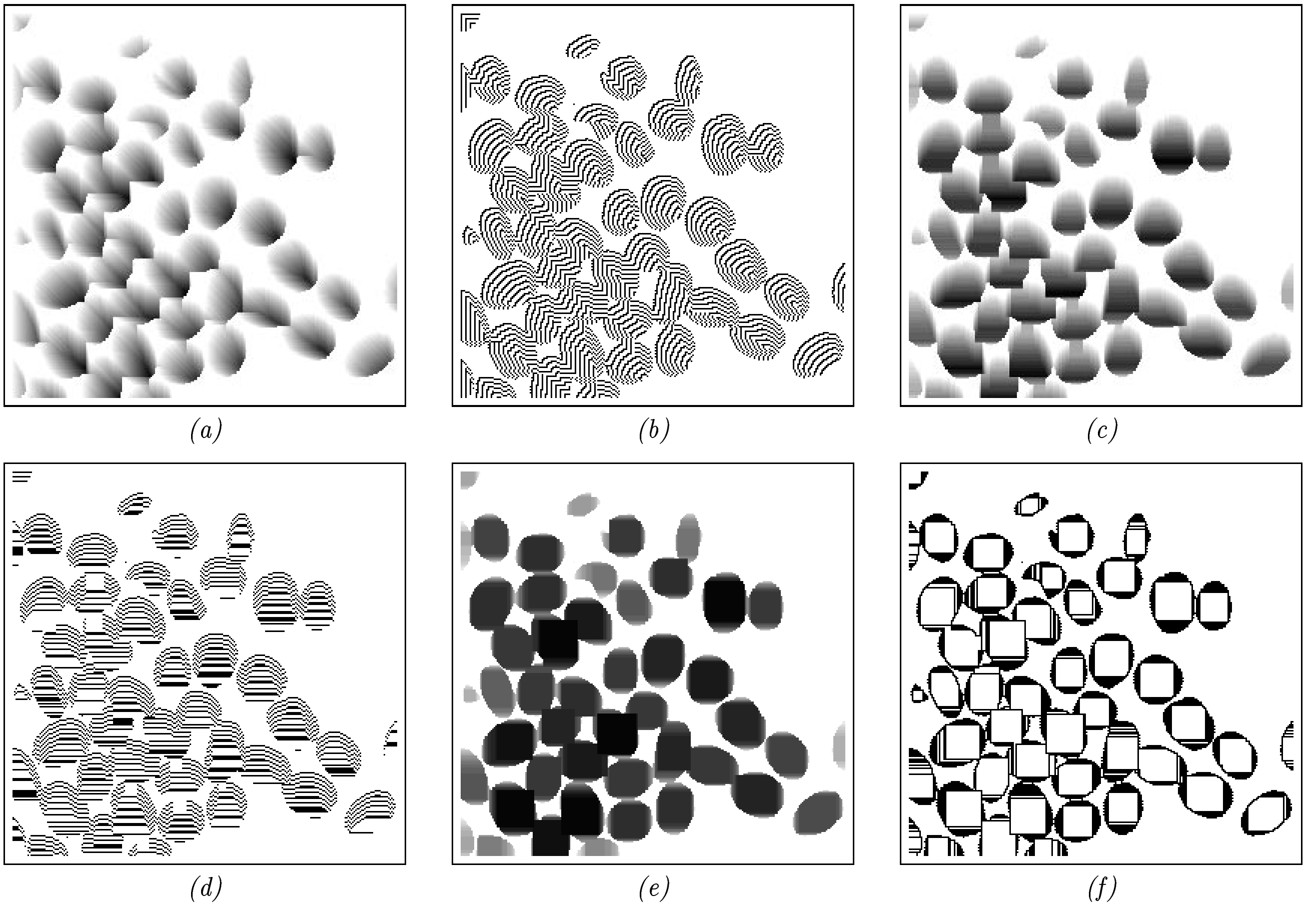


Figure 6 : *Computation of granulometry function using square structuring elements. (a) generalized distance function; (b) level lines; (c) propagation of values from right to left; (d) level lines; (e) final granulometry function; (f) level lines of granulometry function.*

In square grids however, the elementary "diamond" structuring element D *cannot* be decomposed as Minkowski addition of elementary line segments. The closest "approximation" is obtained with

$$D \quad = \quad \begin{array}{ccc} & \bullet & \\ \bullet & \bullet & \bullet \\ & \large\bullet & \end{array} \quad \neq \quad \begin{array}{cc} & \bullet \\ \large\bullet & \end{array} \oplus \begin{array}{cc} \bullet & \\ & \large\bullet \end{array} \quad = \quad E_3 \quad \oplus \quad E_4 \tag{20}$$

and does not contain the central pixel of D! Therefore, starting from distance function $d_S(X)$ 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 by 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.

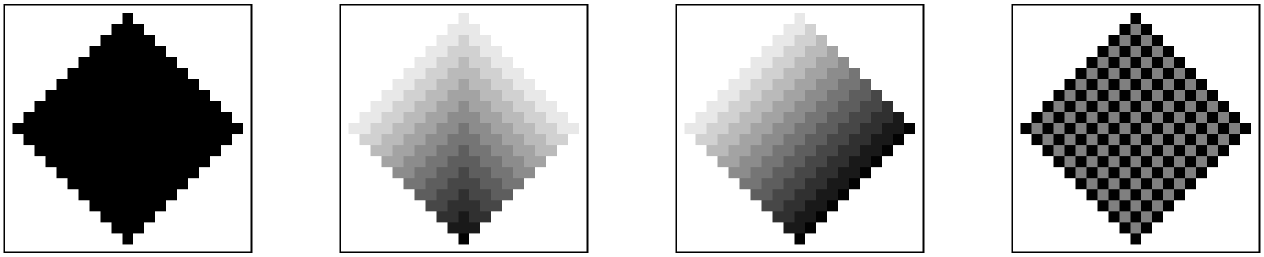

(a) Original shape *(b) Distance function* *(c) SW to NE propagation* *(d) SE to NW propagation*

Figure 7: *When using "diamonds" as structuring elements, the two propagation steps of the granulometry function algorithm need to be followed by a "hole-filling" step.*

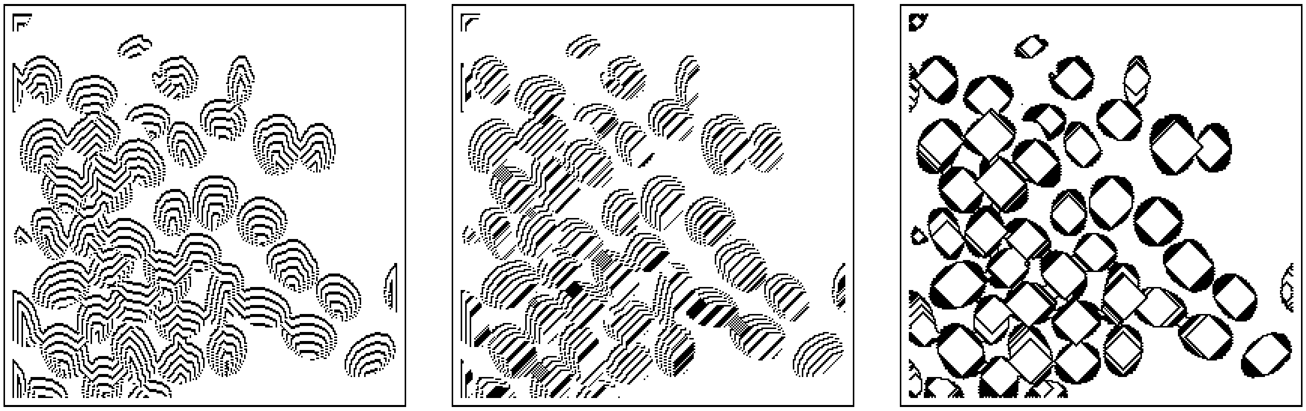

(a) *(b)* *(c)*

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*

Table 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's algorithm [5] 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 proposed 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.*

5 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. [7]); however, since until today, 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 introduce a new algorithm for computing linear grayscale granulometries. A follow-up to this paper describes this algorithm in greater detail, and also deals with the case of truly 2-D grayscale granulometries [18].

Like in section 3, let us for example deal with the horizontal case. The structuring elements considered are the L_n's of equation (8). Let I be a discrete grayscale image.

Definition 11 *A* horizontal maximum *M of length $l(M) = n$ in grayscale image I is a set of pixels $\{p, N_0^{(1)}(p), N_0^{(2)}(p), \ldots, N_0^{(n-1)}(p)\}$ such that*

$$\forall i,\ 0 < i < n, I(N_0^{(i)}(p)) = I(p) \quad \text{and} \quad I(N_4(p)) < I(p), \quad I(N_0^{(n)}(p)) < I(p). \tag{21}$$

The study of how such maxima are altered through horizontal openings is at the basis of the algorithm introduced here. The following proposition holds:

Proposition 12 *Let M be a horizontal maximum of I. Let $p_l \in M$ 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:*

$$\forall k < n, \forall p \in M, \qquad (I \circ L_k)(p) = I(p), \tag{22}$$

$$\text{for } k = n, \forall p \in M, \qquad (I \circ L_n)(p) = \max\{I(N_4(p_l)), I(N_0(p_r))\} < I(p), \tag{23}$$

$$\forall k > n, \forall p \in M, \qquad (I \circ L_k)(p) < I(p). \tag{24}$$

Intuitively, this means that any opening of I by a line segment L_k such that $k < n$ leaves this maximum unchanged, whereas for any $k >= n$, 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 $\max\{I(N_4(p_l)), I(N_0(p_r))\}$. In granulometric terms, the contribution of maximum M to the n's bin of the horizontal pattern spectrum $\mathrm{PS}_0(I)$ is:

$$n \times [I(p) - \max\{I(N_4(p_l)), I(N_0(p_r))\}].$$

This is illustrated by Fig. 9.

In addition, the effect of the horizontal opening of size n on M results in a new “plateau” of pixels being created at altitude $\max\{I(N_4(p_l)), I(N_0(p_r))\}$. This plateau may or may not be itself a maximum of $I \circ L_n$.

Further to these remarks, the principle of the introduced grayscale granulometry algorithm is to scan the lines of I one after the other. Each horizontal maximum M of the current line is identified, and its contribution to $\mathrm{PS}_0(I)(l(M))$ 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

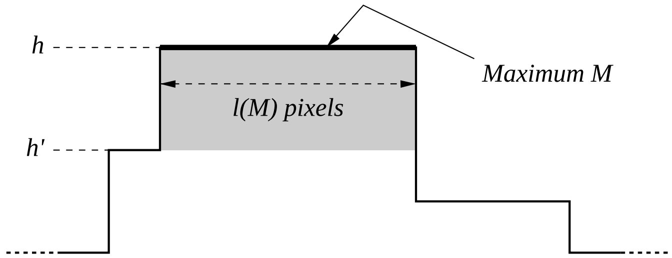


Figure 9 : *Horizontal cross section of I with a maximum M. The shaded area, of volume $(h-h') \times l(M)$ shows the local contribution of this maximum to the $l(M)$-th bin of the horizontal pattern spectrum.*

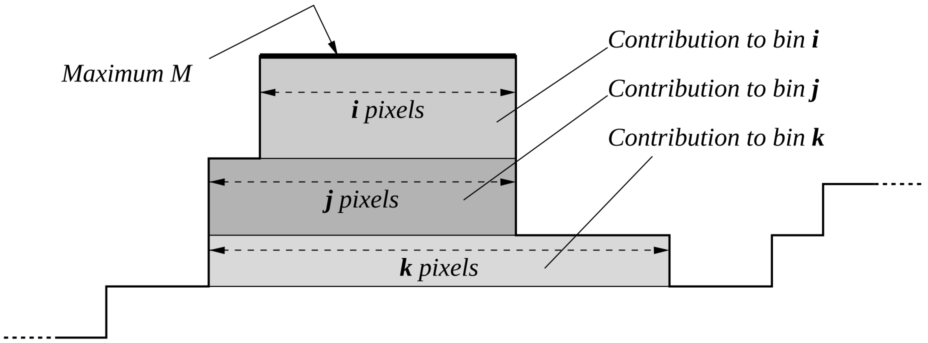


Figure 10 : *How the contribution of the "maximal region" surrounding a maximum M is determined in the computation of the pattern spectrum.*

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. 10.

At any time during this process, the left and right pixels of each "maximal region" processed are stored, so that this region does not have to be scanned again later. The resulting algorithm is therefore linear with respect to the number of pixels in the image, and in the worst case, each image pixel is only scanned *twice.* In practice, the execution time varies only slightly from image to image, depending on the number and complexity of maximal regions found in each scanline.

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. 11 were used for this comparison. As illustrated by table 2, the new algorithm introduced in this paper is up to three orders of magnitude faster!

	Traditional (size 1 to 512)	Traditional (size 1 to 30)	New algorithm (size 1 to 512)
Weld image (a)	204s	13.0s	0.248s
Weld image (b)	204s	12.7s	0.206s

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

The speed of this new algorithm opens up 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 the algorithm introduced in this section, it becomes possible to use grayscale granulometries more systematically. These tools provide indeed 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. 11a–b are used to illustrate how grayscale granulometries can be used to estimate size information[2]. 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 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 Figs. 11a and in Fig. 11b is of 4 pixels and 12 pixels respectively.

6 Conclusion

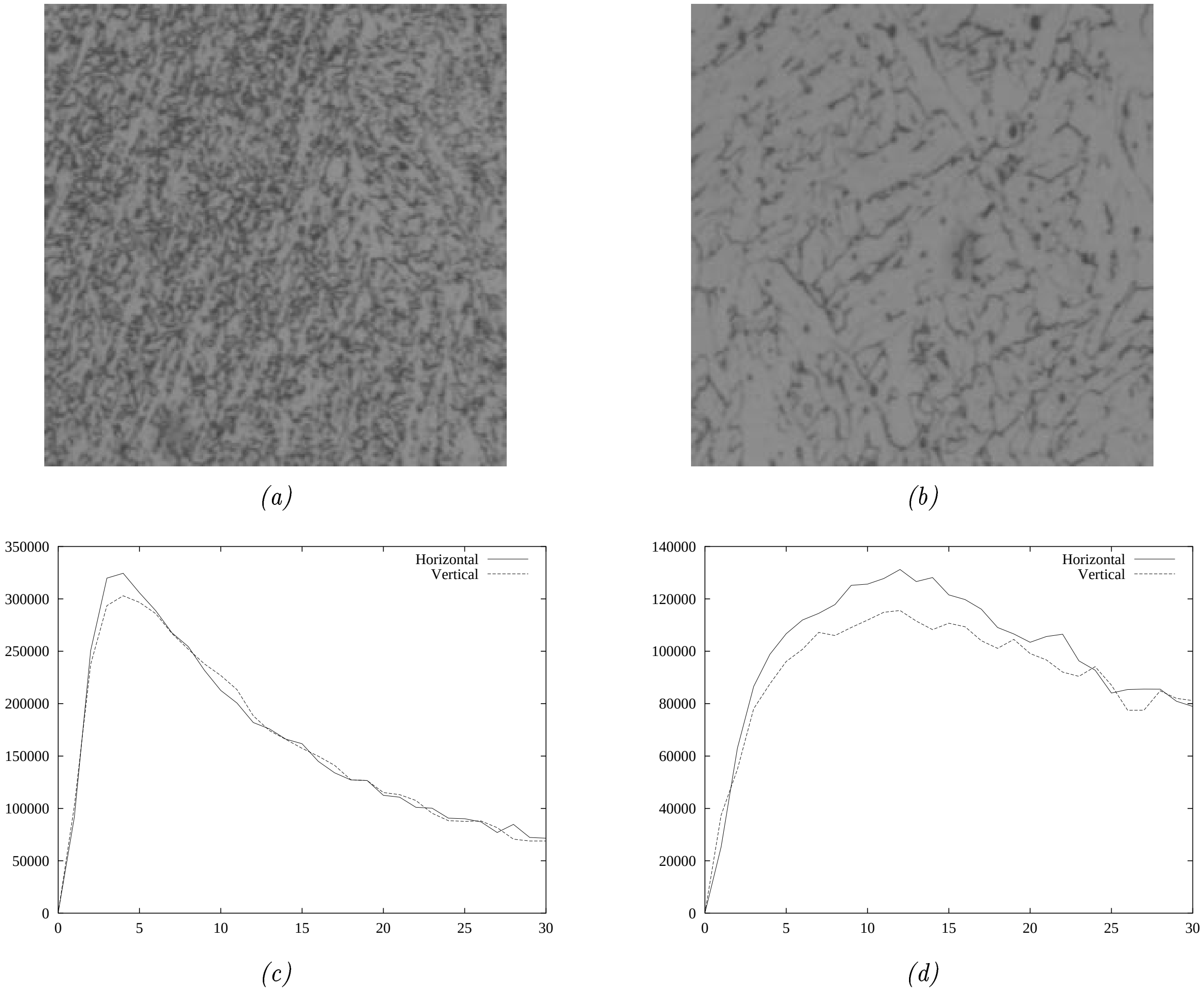
Even though the concept of granulometry was introduced over 25 years ago, the computation time required to extract granulometric curves made it impossible to use these curves for most practical applications. In this paper, a comprehensive set of fast algorithms to compute granulometries in binary images was first described. Most of these algorithms use opening functions as an intermediate step, and they are shown to be faster than any previously available method.

A new algorithm for computing granulometries of grayscale images using openings or closings with linear structuring elements is also introduced. This algorithm is so much faster (up to three orders of magnitude) than any previously available technique that it makes it possible to use grayscale granulometries on a "routine" basis. Examples of application show how these granulometries can be used to extract global size information directly from a grayscale image. These granulometries being one-dimensional, they can also be successfully used on 1-D signals. In [18], this granulometry algorithm is generalized to the true 2-D case. It is expected these new algorithms will greatly contribute to popularize the use of grayscale granulometries in the image and signal analysis community.

7 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*, Boston MA, Nov. 1992.
4. 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.
5. 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.
6. R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*. Addison-Wesley, 1991.
7. E. J. Kraus, H. Heijmans, and E. R. Dougherty. Gray-scale granulometries compatible with spatial scalings. *Signal Processing*, 34:1–17, 1993.

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



(a) *(b)*

(c) *(d)*

Figure 11 : *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.*

8. 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.

9. P. Maragos. Pattern spectrum and multiscale shape representation. *IEEE Trans. Pattern Anal. Machine Intell.*, 11(7):701–716, July 1989.

10. G. Matheron. *Eléments pour une Théorie des Milieux Poreux.* Masson, Paris, 1967.

11. G. Matheron. *Random Sets and Integral Geometry.* John Wiley and Sons, New York, 1975.

12. A. Rosenfeld and J. Pfaltz. Sequential operations in digital picture processing. *J. Assoc. Comp. Mach.*, 13(4):471–494, 1966.

13. A. Rosenfeld and J. Pfaltz. Distance functions on digital pictures. *Pattern Recognition*, 1:33–61, 1968.

14. M. Schmitt. Variations on a theme in binary mathematical morphology. *Journal of Visual Communication and Image Representation*, 2(3):244–258, Sept. 1991.

15. J. Serra. *Image Analysis and Mathematical Morphology.* Academic Press, London, 1982.

16. L. Vincent. Efficient computation of various types of skeletons. In *SPIE Vol. 1445, Medical Imaging V*, pages 297–311, San Jose, CA, 1991.

17. L. Vincent. New trends in morphological algorithms. In *SPIE/SPSE Vol. 1451, Nonlinear Image Processing II*, pages 158–169, San Jose, CA, Feb. 1991.

18. 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.

19. L. Vincent and S. Beucher. The morphological approach to segmentation: an introduction. Technical report, Ecole des Mines, CMM, Paris, 1989.

20. 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.

21. 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.