FAST GRAYSCALE GRANULOMETRY ALGORITHMS

LUC VINCENT
Xerox Imaging Systems
9 Centennial Drive, Peabody, MA 01960, USA

[Proc. ISMM'94 International Symposium on Mathematical Morphology, *Fontainebleau, France, pp. 265–272, September 1994,* Kluwer Academic Publishers*]*

Abstract.
Granulometries constitute an extremely useful set of morphological operators, applicable to a variety of image analysis tasks. Traditional granulometry algorithms involve sequences of openings or closings of increasing size, and are therefore very slow on non-dedicated hardware. Efficient techniques have been proposed to compute granulometries in binary images, based on the concept of *opening functions.* In the present paper, a class of algorithms for computing granulometries in grayscale images is introduced. The most advanced among them are based on the new concept of *opening tree.* These algorithms are several orders of magnitude faster than traditional techniques, thereby opening up a range of new applications for grayscale granulometries.

Key words: Algorithms, Mathematical Morphology, Grayscale Granulometries, Pattern Spectrum, Opening Tree

1. Introduction, State of the Art

In a variety of image analysis problems, one is interested in extracting the *size distribution* of the "objects" or "structures" present in an image. In 1967, Matheron formally characterized meaningful size distributions by introducing the concept of *granulometry* [9, 10]:

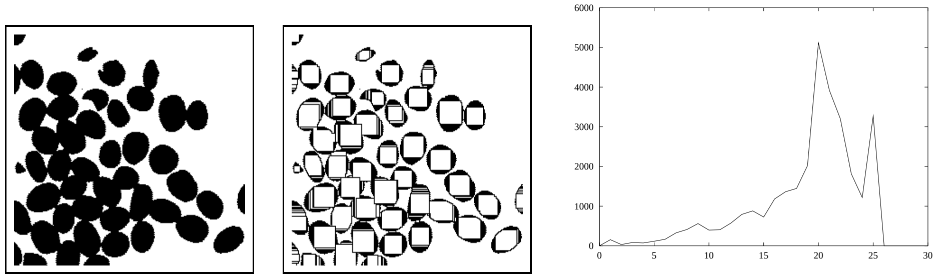
Definition 1 *A granulometry is a family of openings $\Phi = (\phi_\lambda)_{\lambda \geq 0}$ satisfying:*

$$\forall \lambda \geq 0, \mu \geq 0, \quad \lambda \geq \mu \Longrightarrow \phi_\lambda \leq \phi_\mu. \tag{1}$$

Performing the granulometric analysis of an image I with Φ consists in mapping each size λ to a measure of the opened image $\phi_\lambda(I)$. Similarly, anti-granulometries, or granulometries *by closings,* can be defined as families of increasing closings. For more details on granulometries, see [10].

In the discrete case, a granulometry is a decreasing family of openings $\Phi = (\phi_n)_{n \geq 0}$, indexed on an integer parameter n. Denote by $m(I)$ the measure of a discrete image I: $m(I)$ is the area of I in the binary case[1] (number of "on" pixels) and the volume of I in the grayscale case (sum of pixel values). The granulometric analysis of I with Φ results in a granulometric curve:

[1] In the binary case, it may be useful for some granulometries to define $m(I)$ as the number of connected components in I [2].



(a) (b) (c)

Fig. 1. (a) original binary image of coffee beans, (b) level lines of "square" opening function, (c) pattern spectrum obtained as histogram of the opening function.

Definition 2 *The granulometric curve or* pattern spectrum [8] *of an image, I, with respect to granulometry $\Phi = (\phi_n)_{n \geq 0}$ is the mapping $PS_\Phi(I)$ given by:*

$$\forall n > 0, \quad PS_\Phi(I)(n) = m(\phi_n(I)) - m(\phi_{n-1}(I)). \tag{2}$$

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.

Until recently, determining a granulometric curve involved computing a sequence of openings of increasing size, which is very time consuming no matter how efficient the opening algorithm. Even when $\phi_n(I)$ (n-th opening in the series) can be computed in constant time (for a fixed image size), determining the pattern spectrum using openings of size 1 through n is still an $O(n)$ algorithm. For most applications, this is prohibitively costly, unless specialized hardware is used for computing the successive openings.

Over the past few years, a number of algorithms have been proposed for fast granulometries in binary images (see [6, 14, 5] and more recently [12]). These algorithms typically involve as an intermediate step the extraction of the "opening" function of the binary image studied [12]: this function maps each pixel with the size of the first opening that removes it from the image. It can be computed efficiently for all classic structuring element shapes, and the histogram of the resulting grayscale image provides the desired pattern spectrum. An example is shown in Fig. 1.

This paper treats the case of grayscale granulometries, which has not been previously addressed in literature. In section 2, a fast linear grayscale granulometry algorithm is first described (this algorithm was already briefly mentioned in [12]). Then, in section 3, the concept of *opening trees* is introduced as a way to compactly encode all the successive linear openings of a grayscale image. This concept is at the basis of another set of new algorithms for the fast computation of granulometries with maxima of linear openings, as well as approximations of square granulometric curves. Timing results prove that the introduced algorithms are several orders of magnitude faster than traditional techniques.

2. Linear Granulometries

In this section, the case of granulometries using openings with line segments is considered[2]. For simplicily, let us assume that the line segments used as structuring elements are horizontal (the algorithm easily extends to any orientation). The opening ϕ_n in the series is the opening with structuring element L_n:

$$L_n \quad = \quad \underbrace{\bullet\ \ \bullet\ \ \bullet \cdots \bullet\ \ \bullet}_{n+1 \text{ pixels}} \tag{3}$$

Let us 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.

Definition 3 *A horizontal line segment S, of length $l(S)$, is a set of pixels $\{p_0, p_1, \ldots, p_{n-1}\}$ such that for $0 < i < n$, $p_i = N_r(p_{i-1})$.*

Definition 4 *A* horizontal maximum *M of length $l(M) = n$ in grayscale image I is a horizontal line segment $\{p_0, p_1, \ldots, p_{n-1}\}$ such that:*

$$\forall i, 0 < i < n, \;\; I(p_i) = I(p_0) \quad \textit{and} \quad I(N_l(p_0)) < I(p_0), \;\; I(N_r(p_{n-1})) < I(p_0). \tag{4}$$

This notion is the 1-D equivalent of the classic *regional maximum* concept [7]. The study of how such maxima are altered through horizontal openings is at the basis of the algorithm introduced in this section. Denote by $I \circ B$ the opening of image I by a structuring element B. The following proposition holds:

Proposition 5 *Let $M = \{p_0, p_1, \ldots, p_{n-1}\}$ be a horizontal maximum of I.*

$$\forall k < n, \forall p \in M, \quad (I \circ L_k)(p) = I(p), \tag{5}$$

$$\textit{for } k = n, \forall p \in M, \quad (I \circ L_n)(p) = \max\{I(N_l(p_0)), I(N_r(p_{n-1}))\} < I(p). \tag{6}$$

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, the contribution of maximum M to the n-th bin of the horizontal pattern spectrum $\mathrm{PS}_h(I)$ is:

$$n \times [I(p) - \max\{I(N_l(p_0)), I(N_r(p_{n-1}))\}]. \tag{7}$$

This is illustrated by Fig. 2.

Additionally, the local effect of a horizontal opening of size n on M is that a new "plateau" of pixels is created at altitude $\max\{I(N_l(p_0)), I(N_r(p_{n-1}))\}$. This plateau P contains M, and may be itself a maximum of $I \circ L_n$. If it is, we say that P is part of the *maximal region* $R(M)$ surrounding maximum M, and we can now compute the contribution of P to the $l(P)$-th bin of the pattern spectrum, etc. Following this principle, the proposed granulometry algorithm works as follows on each image line:

[2] The algorithm described in this section applies to 1-D signals as well.

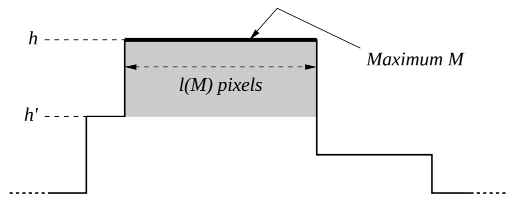


Fig. 2. Horizontal cross section of I with a maximum M. The shaded area, of volume $(h-h')\times l(M)$ shows the local contribution of this maximum to the $l(M)$-th bin of the horizontal pattern spectrum.

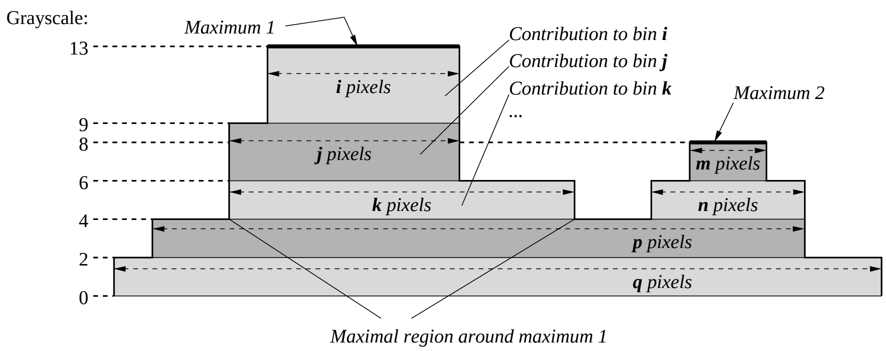


Fig. 3. Illustration of horizontal grayscale granulometry algorithm for a line with two maxima. In this case, maxima were scanned from left to right. Processing maximum 1 results in bins i, j, and k of the pattern spectrum being incremented. While processing maximum 2, bins m and n are first incremented, then the algorithm skips over the already processed maximal region, and bins p and q are incremented. The maximal region around maximum 2 is the entire line.

Algorithm: horizontal granulometry of an image line

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

Note that this algorithm is inherently recursive: once a maximal region R has been processed, all of its pixels are regarded as having the gray-level they were given by the last opening considered for R. In practice though, there is no need to physically modify the values of all the pixels in R: keeping track of the first and last pixels of R is sufficient, allowing the algorithm to efficiently skip over already processed maximal regions (see Fig. 3). 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 I, the new algorithm described in this section is *three orders of magnitude faster.*

3. Granulometries with Maxima of Linear Openings

In order to deal with more complicated cases, we now introduce a technique that can be seen as a generalization of the concept of opening functions for the grayscale case. 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 opening function at pixel p. In other words, the opening function encodes for each pixel the successive values it takes for increasing opening sizes (namely, a series of of 1's followed by a series of 0's). Similarly, in the grayscale case, as the size of the opening increases, the value of each pixel decreases monotonically. If each pixel was assigned the list of values it takes for every opening size, then the corresponding grayscale granulometry could be extracted straightforwardly.

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

Let M be a horizontal maximum of image I, with altitude (grayscale) h. We pointed out in the previous section that a horizontal opening of size $l(M)$ of M takes all of its pixels down to a new value h'. Beyond this, the following proposition can easily be proved:

Proposition 6 *Let $n > 0$, I a grayscale image such that $I = I \circ L_{n-1}$. Then, for every pixel p in I:*

$$(I \circ L_n)(p) < I(p) \iff \exists M \text{ horiz maximum such that } l(M) = n \text{ and } p \in M. \quad (8)$$

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

Based on this principle, the algorithm introduced here represents each image line as a tree T, which we call its *opening tree.* The leaves of T are the image pixels, and the nodes are made of pairs (h, n), where h is a grayscale value and n is an opening size. Every pixel that can be reached by going upwards in the tree starting from node (h, n) is such that its value for the opening of size n is h. Conversely, starting from a pixel p, successive pairs (h_1, n_1), (h_2, v_2),..., (h_1, v_i),..., are reached by going down towards the root of the tree. By convention, for this pixel, $(h_0, n_0) = (I(p), 0)$. These pairs satisfy:

$$\forall i > 0, \quad h_i > h_{i+1} \text{ and } n_i < n_{i+1}. \quad (9)$$

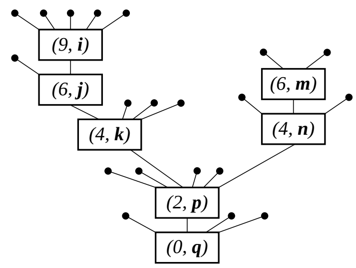


Fig. 4. Opening tree representation of the image line of Fig. 3. The leaves of this tree (•) correspond to the image pixels.

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

$$(I \circ L_n)(p) = h_j, \quad \text{where } j \text{ is such that } n_j \leq n < n_{j+1}. \tag{10}$$

Opening trees can be computed using an algorithm very similar to the one described in the previous section. An example of opening tree is shown in Fig. 4.

Opening trees provide a hierarchical description that can be used to compactly represent *all* the horizontal openings of a grayscale image. In this respect, this notion is a grayscale equivalent of the opening function mentioned earlier (see Fig. 1). One can prove that, in the worst case, the opening tree has one node per image pixel. In practice though, only between 0.3 and 0.9 nodes per pixel are needed depending on the complexity of the image processed.

Any horizontal opening of I can be straightforwardly derived from its opening tree. In addition, the horizontal pattern spectrum of I, $\mathrm{PS}_h(I)$, can be computed from T as follows:

Algorithm: horizontal granulometry of I from its opening tree

- `initialize each bin of pattern spectrum` $\mathtt{PS}_h(I)$ `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:`
    $\mathtt{PS}_h(I)(n) \leftarrow \mathtt{PS}_h(I)(n) + (v - h)$`;`
    $v \leftarrow h$`;` $(h, n) \leftarrow$ `next node down in tree;`

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

Algorithm: granulometry of I from opening trees T_1 and T_2

- `initialize each bin of pattern spectrum` $\mathtt{PS}(I)$ `to 0;`
- `for each pixel` p `of` I `do:`
  - $v \leftarrow I(p)$`;`

- $(h_1, n_1) \leftarrow$ `node of` T_1 `pointed at by` p;
- $(h_2, n_2) \leftarrow$ `node of` T_2 `pointed at by` p;
- `while` (h_1, n_1) `and` (h_2, n_2) `exist, do:`

    `size` $\leftarrow \max(n_1, n_2)$;
    `while` $n_1 \leq$ `size and` (h_1, n_1) `exists do:`
        $(h_1, n_1) \leftarrow$ `next node down in` T_1;
    `while` $n_2 \leq$ `size and` (h_2, n_2) `exists do:`
        $(h_2, n_2) \leftarrow$ `next node down in` T_2;
    `PS`(I)`(size)` $\leftarrow$ `PS`(I)`(size)` $+ (v - \max(h_1, h_2))$;
    $v \leftarrow \max(h_1, h_2)$;

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

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

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

The algorithm given above can be easily adjusted to the computation of *pseudo-granulometries* by *minima* of linear openings (for more details, see [13]). Although minima of openings are not themselves openings [10], minima of openings with line segments of increasing length constitute a decreasing family of image operators. The resulting pseudo-granulometric curves often characterize the same image features as square granulometries[3]. They have been very successful as one of the feature sets used to characterize plankton in towed video microscopy images [3].

4. Conclusion

In the present paper, a novel class of algorithms was introduced for efficiently computing granulometries in grayscale images. These algorithms turn out to be several orders of magnitude faster than any previously available technique. One of their key underlying concepts is that of *opening tree:* such structures are shown to provide a compact representation for the successive openings of a grayscale image by

[3] The author wishes to thank Gaile Gordon for this useful insight.

line segments of increasing size. They are at the heart 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. These algorithms can even be extended to the fast computation of granulometries by area openings and closings [11], as described in [13]. However, the efficient computation of *exact* grayscale granulometries with square openings remains an open problem (no pun intended).

Over the past few years, grayscale granulometries have proved to be useful in a variety of image analysis tasks, including texture classification [1] and picture segmentation [4]. However, these applications have often been handicapped by slow processing speeds. The efficiency of the algorithms introduced in the present paper not only makes it now possible to use granulometries on a routine basis, they also open new areas of application for these tools. For example, in [12], it was briefly shown how linear granulometries can be used to directly extract *global* estimates of object size in grayscale images without the need for any segmentation or binarization. Extracting such information is sometimes a goal in itself, but it can also be essential to calibrate the parameters of subsequent image analysis algorithms to be applied, thereby greatly enhancing their robustness. Undoubtedly, the family of algorithms introduced in this paper will contribute to popularize the use of granulometries for a growing number of image and signal analysis problems.

References

1. 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.
2. M. Coster and J.-L. Chermant. *Précis d'Analyse d'Images*. CNRS Ed., Paris, 1985.
3. 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.
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. 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.
7. F. Maisonneuve. Extrema régionaux: Algorithme parallèle. Technical Report 781, Ecole des Mines, CGMM, Paris, 1982.
8. P. Maragos. Pattern spectrum and multiscale shape representation. *IEEE Trans. Pattern Anal. Machine Intell.*, 11(7):701–716, July 1989.
9. G. Matheron. *Eléments pour une Théorie des Milieux Poreux*. Masson, Paris, 1967.
10. G. Matheron. *Random Sets and Integral Geometry*. John Wiley and Sons, New York, 1975.
11. L. Vincent. Morphological area openings and closings for grayscale images. In *NATO Shape in Picture Workshop*, pages 197–208, Driebergen, The Netherlands, Sept. 1992.
12. L. Vincent. Fast opening functions and morphological granulometries. In *SPIE Vol. 2300, Image Algebra and Morphological Image Processing V*, pages 253–267, San Diego, CA, July 1994.
13. L. Vincent. Grayscale granulometries: Fast algorithms and new image analysis applications. manuscript, to be published, June 1994.
14. 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.