Proc. EURASIP Workshop on Mathematical Morphology and its Applications to Signal Processing, Barcelona, Spain, pp. 22–27, May 1993.

Grayscale area openings and closings, their efficient implementation and applications

Luc Vincent

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

Abstract

The filter that removes from a binary image its connected components with area smaller than a parameter λ is called *area opening.* From a morphological perspective, this filter is an algebraic opening, and it can be extended to grayscale images. The properties of area openings and their dual area closings are recalled. In particular, it was proved in [13] that the area opening of parameter λ of an image I is the supremum of the grayscale images that are smaller than I and whose regional maxima are of area greater than or equal to λ. This theorem is at the basis of an efficient algorithm for computing grayscale area openings and closings. Its implementation involves scanning pixels in an order that depends both on their location and value. For this purpose, the use of *pixel heaps* is proposed. This data structure is shown to be both efficient and low in memory requirements. In addition, it can be used in the computation of various other complex morphological transforms. The use of these area openings and closings is illustrated on image filtering and segmentation tasks

1 Notations, Definitions

In this paper, the binary images or sets under study are subsets of a connected compact set $\mathbf{M} \subset \mathbb{R}^2$ called the *mask.* Similarly, grayscale images are mappings from $\mathbf{M}$ to $\overline{\mathbb{R}}$. For simplicity, only the 2-D case is considered here, but the notions and algorithms discussed generalize to arbitrary dimensions. By area is meant the *Lebesgue* measure in $\mathbb{R}^2$. γ_B denotes the morphological opening with respect to structuring element B.

Let us first recall the notion of connected openings [9, 10]:

Definition 1 *The connected opening $C_x(X)$ of a set $X \subseteq \mathbf{M}$ at point $x \in \mathbf{M}$ is the connected component of X containing x if $x \in X$ and $\emptyset$ otherwise.*

The area opening γ^a_λ can then be defined on subsets of $\mathbf{M}$ as follows [13]:

Definition 2 (binary area opening) *Let $X \subset \mathbf{M}$ and $\lambda \geq 0$. The area opening of parameter λ of X is given by*

$$\gamma^a_\lambda(X) = \{x \in X \mid \mathit{Area}(C_x(X)) \geq \lambda\}. \qquad (1)$$

Intuitively, if $(X_i)_{i \in I}$ denote the connected components of X, $\gamma^a_\lambda(X)$ is equal to the union of the X_i's with area greater than or equal to λ:

$$\gamma^a_\lambda(X) = \bigcup\{X_i \mid i \in I, \mathit{Area}(X_i) \geq \lambda\}. \qquad (2)$$

An example of binary area opening is shown in Fig. 1. Note on this image that, by definition, the connected components of the original image are either removed, or entirely preserved.

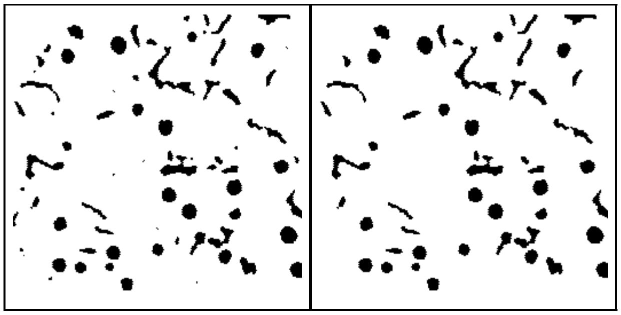


Figure 1: (a) original image; (b) binary area opening.

Obviously, γ^a_λ is *increasing, idempotent,* and *anti-extensive.* It is therefore an algebraic opening [6, 9, 10]. Its dual binary area closing can be defined as follows:

Definition 3 *The area closing of parameter $\lambda \geq 0$ of $X \subset \mathbf{M}$ is given by:*

$$\phi^a_\lambda(X) = [\gamma^a_\lambda(X^C)]^C.$$

where X^C denotes the complement of X in $\mathbf{M}$, i.e. the set $\mathbf{M} \setminus X$ ($\setminus$ denoting the set difference operator). As the dual of the area opening, the area closing fills in the holes of X whose area is strictly smaller than the size parameter λ.

As shown in [13], the growth of these transformations makes it possible to extend them to grayscale images:

Definition 4 (grayscale area opening) *For a mapping $f : \mathbf{M} \longrightarrow \overline{\mathbb{R}}$, the area opening $\gamma_\lambda^a(f)$ is given by:*

$$(\gamma_\lambda^a(f))(x) = \sup\{h \leq f(x) \mid x \in \gamma_\lambda^a(T_h(f))\}, \quad (3)$$

or:

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

In this definition, $T_h(f)$ stands for the threshold of f at value h, i.e:

$$T_h(f) = \{x \in \mathbf{M} \mid f(x) \geq h\}. \quad (4)$$

In other words, to compute the area opening of f, the area openings $\gamma_\lambda^a(T_h(f))$ of the thresholds $T_h(f)$ of f are considered. Since γ_λ^a is increasing, $Y \subseteq X \implies \gamma_\lambda^a(Y) \subseteq \gamma_\lambda^a(X)$. Thus, the $\{\gamma_\lambda^a(T_h(f))\}_{h \in \mathbb{R}}$ are a decreasing sequence of sets which by definition constitute the threshold sets of the transformed mapping $\gamma_\lambda^a(f)$.

By duality, one similarly extends the concept of area closing to mappings from **M** to $\overline{\mathbb{R}}$. These area openings and closings for grayscale images are typical examples of flat increasing mappings (also called stack mappings) [10]. Their geometric interpretation is relatively simple: unlike dynamics-based openings [2], which remove structures based on their *contrast*, grayscale area openings remove from an image all the light structures that are "smaller" than the size parameter, i.e., based on their *area* (number of pixels). Area closings have the same effect on dark structures. Theorem 8 below provides a more refined interpretation of this intuitive interpretation.

2 Properties

In this section, a completely different interpretation of area openings and closings is given. For more details, including proofs, refer to [13].

A well-known theorem by Matheron states that a translation-invariant algebraic opening γ is the supremum of all the morphological openings γ_B that are smaller than or equal to γ [6]. In the particular case of area openings, a more precise characterization of these morphological openings can be given:

Theorem 5 *Denoting by $\mathcal{A}_\lambda$ the class of the subsets of* **M** *which are connected and whose area is greater than or equal to λ, the following equation holds:*

$$\gamma_\lambda^a = \bigcup_{B \in \mathcal{A}_\lambda} \gamma_B. \quad (5)$$

Similarly, it can be proved that the area closing of parameter λ is equal to the infimum of all the closings with connected structuring elements of area greater than or equal to λ.

In the discrete domain, any connected set of area greater than or equal to $\lambda \in \mathbb{N}$ contains a connected set of area equal to λ. The theorem can thus be made more specific as follows:

Corollary 1 *Let $\mathbb{Z}^2$ be the discrete plane equipped with e.g., 4- or 8-connectivity. For $X \in \mathbb{Z}^2 \cap \mathbf{M}$ and $\lambda \in \mathbb{N}$,*

$$\gamma_\lambda^a(X) = \bigcup\{\gamma_B(X) \mid B \in \mathbb{Z}^2 \textit{ connected }, \mathit{Area}(B) = \lambda\}.$$

Theorem 5 is easily extended to grayscale as follow:

Proposition 6 *Let $f : \mathbf{M} \longrightarrow \overline{\mathbb{R}}$, be an upper semi-continuous mapping [8, pp. 425–429]. The area opening of f is given by:*

$$\gamma_\lambda^a(f) = \bigvee_{S \in \mathcal{A}_\lambda} \gamma_S(f). \quad (6)$$

A dual proposition can be stated for grayscale area closings.

The previous proposition leads to a different understanding of area openings (respectively closings). As a maximum of openings with all possible connected elements of area greater than or equal to a given λ, it can be seen as *adaptive:* at every location, the (connected) structuring element adapts its shape [1] to the image structure so as to "remove as little as possible" (see Fig. 2).

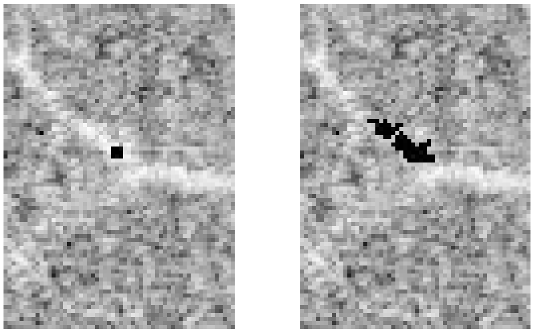


Figure 2: Local shape of an"adaptive structuring element" when centered at the pixel shown in black in the left image.

3 Relation with Regional Extrema

A third and more geometric interpretation of area openings is provided in this section (only openings are dealt with here, the dual case of the closings being easy to derive). Theorem 8 below is at the basis of the algorithm proposed in section 4, so its proof will be given. Let usfirst recall the notion of *maximum* on a mapping [8, page 445].

Definition 7 *Let f be an upper semi-continuous (u.s.c.) mapping from* $\mathbf{M}$ *to* $\overline{\mathrm{I\!R}}$*. A (regional) maximum of f at level $h \in \overline{\mathrm{I\!R}}$ is a connected component M of $T_h(f)$ such that*

$$\forall h' > h,\ T_{h'} \cap M = \emptyset. \tag{7}$$

The following theorem can now be stated:

Theorem 8 *Let f be a u.s.c. mapping from* $\mathbf{M}$ *to* $\overline{\mathrm{I\!R}}$*, $\lambda \geq 0$. Denoting $\mathcal{M}_\lambda$ the class of the u.s.c. mappings g :* $\mathbf{M} \longmapsto \overline{\mathrm{I\!R}}$ *such that any maximum M of g is of area greater than or equal to λ,*

$$\gamma_\lambda^a(f) = \sup\{g \leq f \mid g \in \mathcal{M}_\lambda\}. \tag{8}$$

Proof: Let $g \in \mathcal{M}_\lambda$, $g \leq f$, and let $h \in \overline{\mathrm{I\!R}}$. Let A be an arbitrary connected component of $T_h(g)$. Since g is u.s.c., A is a compact set and therefore, there exists $x \in A$ such that $g(x) = \max\{g(y) \mid y \in A\}$. Let $h' = g(x)$ and $B = C_x(T_{h'}(g))$. B is obviously a maximum of g at altitude h'. Indeed, if there existed a $y \in B$ such that $g(y) > h'$, we would have $y' \notin A$ (the maximal value of g on A is h), and thus $A \subset A \cup B \subseteq T_h(g)$. Furthermore, $A \cup B$ is connected as the union of two connected sets with non-empty intersection, which would be in contradiction with the fact that A is a connected component of $T_h(g)$. B is therefore a maximum at altitude h' of g and $B \subseteq A$. Since by hypothesis, *Area*$(B) \geq \lambda$, we therefore have *Area*$(A) \geq \lambda$.

Thus, for every $h \in \overline{\mathrm{I\!R}}$, $\gamma_\lambda^a(T_h(g)) = T_h(g)$. Besides, $T_h(g) \subseteq T_h(f)$. Therefore, by growth of γ_λ^a, $\gamma_\lambda^a(T_h(g)) = T_h(g) \subseteq \gamma_\lambda^a(T_h(f))$. This being true for every threshold, we conclude that $g \leq \gamma_\lambda^a(f)$.

Conversely, $\forall h \in \overline{\mathrm{I\!R}}$, any connected component A of $T_h(\gamma_\lambda^a(f))$ is of area $\geq \lambda$. Thus, all the maxima of $\gamma_\lambda^a(f)$ are of area $\geq \lambda$. It follows that $\gamma_\lambda^a(f) \in \mathcal{M}_\lambda$ and (anti-extensivity) $\gamma_\lambda^a(f) \leq f$, which completes the proof. □

4 Computation of Area Openings and Closings

Computing area openings in binary images is a straightforward matter: after a labelling of the connected components, the histogram of the image provides the area of each of its components. The too small ones are then removed.

However, things are rather more complicated in the grayscale case:

- Obviously, applying definition 2 and computing $\gamma_\lambda^a(I)$ for every threshold of the original grayscale image I then "piling up" the resulting binary images is a much too computationally expensive operation. Furthermore, its time complexity increases exponentially with the number of bits per pixel. . .
- Similarly, following proposition 6, the computation of all the possible openings with all the possible connected structuring elements with λ pixels becomes an impossible task as soon as λ is greater than 4 or 5. Indeed, the number of possible structuring elements becomes tremendous! Note however that an *approximate* algorithm based on such principles was proposed for $\lambda \leq 8$ [1]. It is however still very slow, inaccurate and the constraint $\lambda \geq 8$ does not leave area openings and closings enough "punch" for most applications.

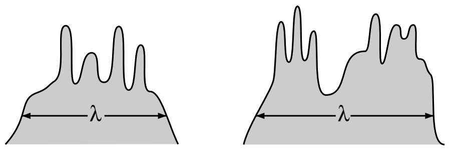
Instead, the algorithm introduced now is based on the third interpretation given for grayscale area openings, namely that formalized in theorem 8. The general principle of the proposed algorithm is to successively consider all the maxima m of the image; for each m, a "local threshold" around it is progressively lowered until its area becomes larger than the parameter λ. In other words, an iso-intensity line is drawn around m and its intensity is decreased until the enclosed region is of area $\geq \lambda$. Denoting by I be the original grayscale image, the successive steps of the proposed algorithm are as follows:

- Extract the regional maxima of I. For this step, refer to [14, 12, 2]).
- For each regional maximum m of I, do the following:
  - If the area of this maximum is larger than λ, go to the next maximum.
  - Recursively scan the neighbors of m, in decreasing order of their gray value, until either of the following two condition is fulfilled:

    ⋆ the number of pixels scanned is larger than $\lambda -$ *Area*(m)

    ⋆ the next pixel on the ordered list of pixels to be visited has a gray value larger than the gray-level of the current pixel.
  - Give all the corresponding pixels of I (including pixels belonging to m) the gray value of the last pixel visited.

In this algorithm, the second condition deals with the case of several (say n) regional maxima of I getting merged into one maximum of J (see Fig. 3a). In such cases, we avoid to scan this new maximum n times by preventing a scanning to proceed when it would lead to other maxima of I. This rule also correctly deals with the case of maxima of small area located next to a large and brighter area. Notice that in order for this method to work, the algorithm has to proceed in a sequential manner [12], i.e., modify the original image after the processing of each maximum! If the maxima of I were to be processed as above, but in a parallel fashion, then the algorithm would have to be iterated until convergence in order to properly deal with hierarchically organized maxima of the kind of Fig. 3b.



(a) *(b)*

Figure 3: (a) set of maxima that get merged into one by area opening; (b) hierarchy of maxima.

The sensitive point of this implementation is the recursive ordered scanning of the neighbors of each maximum. Since the neighboring pixels of the current region have to be processed in decreasing order of gray-level, a simple queue of pixel pointers cannot be used [12]. One has to use a structure which keeps track of the "priority level" (i.e., gray-level) of each pixel. So-called hierarchical queues [7] constitute one approach, but have rather large memory requirements: for an image with M different gray-levels, one would have to allocate as much as M arrays of size λ, which is completely irrealistic for large values of λ and images with, say, 12 significant bits.

Instead, we propose to use data structures that have classically been used in sorting and searching algorithms, namely *priority queues* or *heaps* [3]. A pixel heap is basically a balanced binary tree of pixel pointers which satisfies the heap condition: the grayscale value of any heap pixel is larger than the value of its children. Among other operations that can be performed on a heap, the operations of:

– inserting a new pixel

– removing the pixel with largest value

can be executed in time $O(\log(N))$, N being the number of elements in the heap. These two operations each involve the scanning of at most one full branch of the tree in order to preserve its balanced state as well as the heap condition. An example of insertion of a new element is shown in Fig. 4.

In the present algorithm, the use of a pixel heap is particularly appropriate. Its memory requirements (array of pointers of size λ) are negligible compared to what a hierarchical queue would need! Equally negligible is the speed gain provided by these queues. The heap implementation of the algorithm described above on a *Sun Sparc Station 2* computes area openings of size 100 on a 256 × 256 image in less than 3 seconds on average! Adapting it to area closings is a straightforward task.

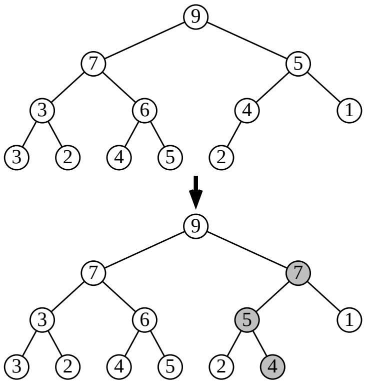
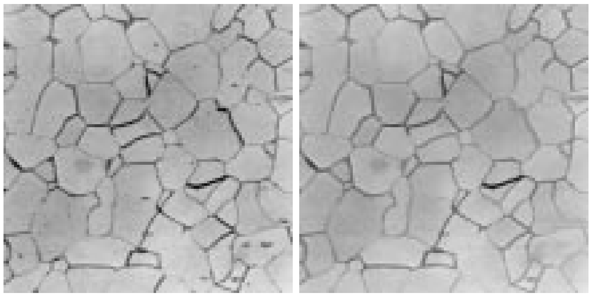


Figure 4: Insertion of a new pixel of value 7 in the pixel heap. Modified nodes are shown in gray.

5 Applications

5.1 Image filtering tasks

It was mentioned earlier that area openings can be seen as morphological openings with structuring elements that adapt their shape to the underlying image structure. This clearly is important for some filtering tasks. Consider for example Fig. 5a, which is a microscopy image of a metallic alloy. It is "corrupted" by some black noise that one may wish to remove. In order to do so without damaging the inter-grain separating lines, a standard method consists of using a minimum of linear closings: one expects that the separating lines are straight enough that they will be preserved at least for one orientation of the linear structuring element. This operation is followed by a dual grayscale reconstruction [4, 14] that reconnects the inter-grains lines that may have been broken by the process. The result is shown in Fig. 5b.



(a) *(b)*

Figure 5: (a) Original image; (b) minimum of linear closings followed by dual grayscale reconstruction.

The problem with this method is that in order to assure a correct preservation of the separating lines, a large number of linear elements with different orientations may be needed. This increases the computational complexity of the algorithm while still requiring that the inter-grains lines be straight enough in places. If these lines wiggle too much, they will also be removed.

Both these speed and accuracy deficiencies can be addressed by using an area closing instead. An area closing of Fig. 5a is shown in Fig. 6a. At first sight, the difference between this image and Fig. 5b is not striking. However, a pixelwise algebraic difference of these images followed by a thresholding extracts the inter-grains zones that have been better preserved by the area closing. This shows that the "adaptive structuring element" of the area closing has played a decisive role. Notice in particular that the better preserved zones are mostly oriented in non vertical, horizontal or diagonal directions. Besides, the computation time of the area closing is far inferior to that of the closing-reconstruction.

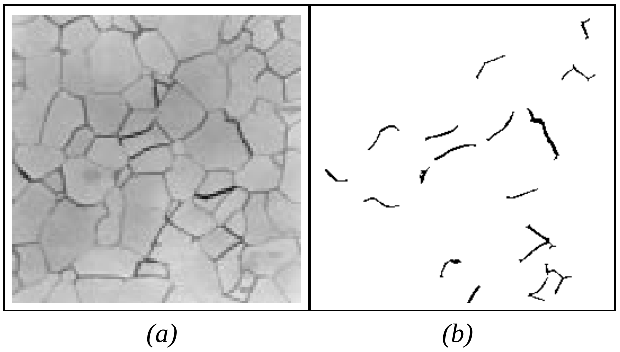


Figure 6: (a) Area closing of Fig. 5a; (b) zones where the area closing outperforms the closing-reconstruction.

Another example of these useful filtering capabilities is shown in Fig. 7[1]. Fig. 7a is a noisy image of fibers, where the black noise needs to be removed while preserving the fibers at best, even the thinnest ones. Once again, the filter that was experimentally found to be best suited for this task is an area closing (see Fig. 7b). It considerably simplifies the image, thus making the automatic extraction of its fibers easier.

In summary, grayscale area openings and closings are particularly suited to filtering tasks where thin and elongated image structures have to be preserved. They can be applied in a relatively systematic fashion, are fast, and usually outperform more standard morphological filtering techniques. Moreover, as described in [13], they can be combined into other kinds of filters, like area Alternating Sequential Filters [8, 13]. In addition, area openings/closings yield filtered images that do not have the "boxy" look sometimes observed when using, e.g., openings/closings with squares: they have nice detail-preserving capabilities.

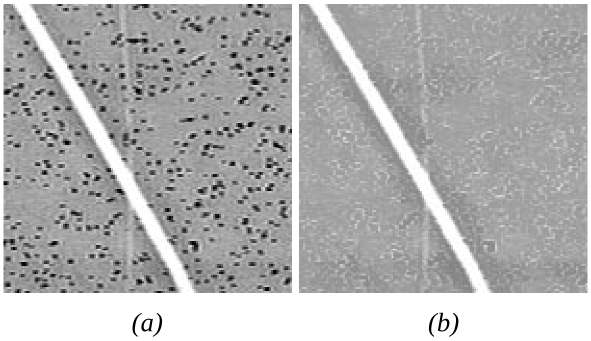


Figure 7: (a) Noisy image of fibers; (b) area closing of (a).

5.2 Use of area openings and closings for image segmentation

Just as with classic openings and closings, one can very well perform top-hats [8] with area openings and closings. This allows the straightforward extraction of small light or dark structures regardless of their shape. As an example, let us consider Fig. 8a, an angiographic image of eye blood vessels where microaneurisms have to be detected. The latter are:

- small and light
- disconnected from the network of the blood vessels,
- predominantly located on the dark areas of the image, i.e. here, the central region.

The algebraic difference between an area opening (of size larger than any possible aneurism) and the original image itself can be called area top-hat and is is shown in Fig. 8b. The aneurisms are clearly visible in this image, and no blood vessel remain.

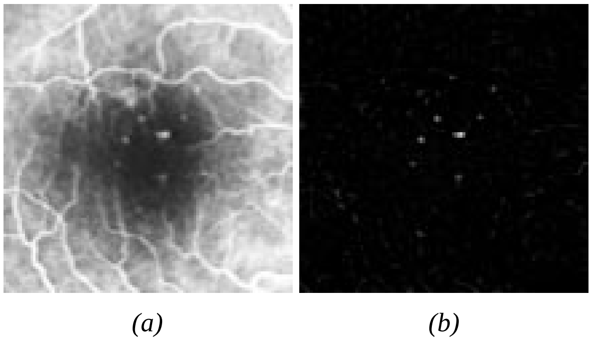


Figure 8: (a) Angiography of eye blood vessels with microaneurisms; (b) area top-hat.

Here, in order to finish the segmentation, more work is necessary: one has to account for the fact that aneurisms are primarily located in dark areas of the image. A solution to this is proposed in [13], and the aneurisms finally extracted are shown in Fig. 9.

[1] Image provided by Hugues Talbot, Ecole des Mines de Paris.

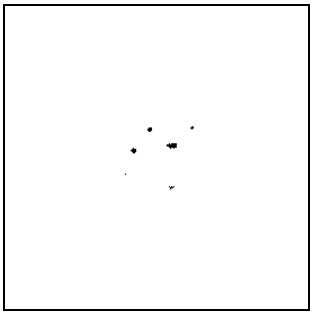


Figure 9: Extracted microaneurisms.

6 Conclusions

In this paper, three different interpretations have been provided for the area opening (resp. closing) of parameter λ of a grayscale image I:

- definition as a flat mapping from the binary area opening
- supremum of all the morphological openings with connected structuring elements of area larger than or equal to the size parameter λ
- supremum of all the grayscale images that are smaller than I and whose regional maxima are of area greater than or equal to λ.

While the first two interpretations do not translate into viable algorithms, the third one leads to a very efficient implementation of these transformations.

The practical use of grayscale area openings and closings was reviewed on a few examples. These operations were shown to be particularly suited to the filtering of noisy images of thin and elongated structures like fibers. Moreover, they can be used for image segmentation via the proposed area top-hat.

Last but not least, the use of a pixel heap for morphological algorithms was introduced. Pixel heaps are both efficient and low in terms of memory requirements. They allow to recursively scan the neighbors of a set of pixels with decreasing (resp. increasing) gray-level, and therefore play a key-role in the proposed area opening algorithm. They also lead to very efficient implementations of a good number of other grayscale morphological algorithms, like:

- watersheds (algorithm derived from [15])
- grayscale reconstruction [14]
- morphological dynamics [2]
- Euclidean skeletons and bisectors [11].

The description of these new methods will constitute the topics of future publication.

References

[1] F. Cheng and A. Venetsanopoulos. Fast, adaptive morphological decomposition for image compression. In *25th Annual Conference on Information Sciences and Systems*, pages 35–40, 1991.

[2] M. Grimaud. A new measure of contrast: Dynamics. In *SPIE Vol. 1769, Image Algebra and Morphological Image Processing III*, pages 292–305, San Diego CA, July 1992.

[3] D. E. Knuth. *The Art of Computer Programming, Vol. 3: Sorting and Searching*. Addison-Wesley, 1973.

[4] C. Lantuéjoul and F. Maisonneuve. Geodesic methods in quantitative image analysis. *Pattern Recognition*, 17(2):177–187, 1984.

[5] P. Maragos and R. Schafer. Morphological filters—part ii: their relations to median, order-statistics, and stack filters. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 35(8):1170–1184, Aug. 1987.

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

[7] F. Meyer. Algorithme ordonné de ligne de partage des eaux. note, Ecole des Mines, CMM, 1990.

[8] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.

[9] J. Serra, editor. *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances*. Academic Press, London, 1988.

[10] J. Serra and L. Vincent. An overview of morphological filtering. *Circuits, Systems and Signal Processing*, 11(1):47–108, Jan. 1992.

[11] H. Talbot and L. Vincent. Euclidean skeletons and conditional bisectors. In *SPIE Vol. 1818, Visual Communications and Image Processing*, pages 862–876, Boston, MA, Nov. 1992.

[12] L. Vincent. Morphological algorithms. In E. R. Dougherty, editor, *Mathematical Morphology in Image Processing*, pages 255–288. Marcel-Dekker, Inc., New York, Sept. 1992.

[13] L. Vincent. Morphological area openings and closings for grayscale images. In *NATO Shape in Picture Workshop*, pages 197–208, Driebergen, The Netherlands, Sept. 1992.

[14] L. Vincent. Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE Transactions on Image Processing*, 2(2):176–201, Apr. 1993.

[15] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Machine Intell.*, 13(6):583–598, June 1991.