Proc. 8th International Congress for Stereology, Irvine CA, August 1991. Also appeared in Acta Stereologica, *Vol. 11/Suppl 1, pp. 521–532, February 1992.*

Recent developments in morphological algorithms

Luc Vincent

Division of Applied Sciences, Harvard University,
Pierce Hall, 29 Oxford St, Cambridge MA 02138, USA.

Abstract

In the implementation of morphological image analysis transformations, several algorithmic techniques can be considered. The well-known parallel and sequential algorithms generally involve a large number of scannings of *all* the pixels and are thus inefficient on classical computers. To overcome this drawback, there have been proposed two main families of algorithms, whose common characteristic is to be based on contours. This allows us to take into account at each step only those pixels whose value may be modified, and results in particularly fast algorithms. The first family was introduced by M. Schmitt and is based on the encoding of the object's boundaries as loops and the propagation of these structures in the image. These methods are the most efficient ones for 2-D binary geodesic operations. In particular, they provide the only known efficient algorithm for computing propagation functions.

The algorithms of the second category, with which we will be mainly concerned, regard images as graphs and realize breadth-first scannings of these graphs by means of queues of pixels. They are almost as fast as the loop algorithms and above all, are especially flexible. In particular, they work equally well in many frameworks: 4-, 6- or 8-connectivity, n-dimensional images, geodesic spaces and even graphs—thereby opening a range of new applications for mathematical morphology. Adapting them from one framework to the other is straightforward. Furthermore, this family of algorithms is suited to the computation of both binary and grayscale operations. Among others, the implementation of such complex transformations as binary and grayscale skeletons, geodesic reconstructions and watersheds can be achieved using these methods. Here, the speed gain with respect to parallel algorithms (running on conventional computers) is of 2 to 3 orders of magnitude

Keywords: algorithm, distance function, First-In-First-Out structure, loop, mathematical morphology, skeleton, watersheds.

Introduction

The purpose of the present paper is to give some hints on the way to efficiently implement morphological transformations on conventional computers. Classical parallel and sequential techniques remain of interest for some operations, and will be recalled in the next section. The main part of the paper is then devoted to the latest families of algorithms introduced in morphology. They are particularly suited to the implementation of complex transformations like propagation functions, watersheds, skeletons. The emphasis is thus put on these operations.

We consider here discrete binary and gray-level images, in square or hexagonal grids. No implementation details is given below. For such technicalities, we refer to (Schmitt, 1989) and (Vincent, 1990). Instead, every category of algorithms is described in an intuitive way: its strongest points are brought to the fore and it is illustrated by some operations which it best allows to determine.

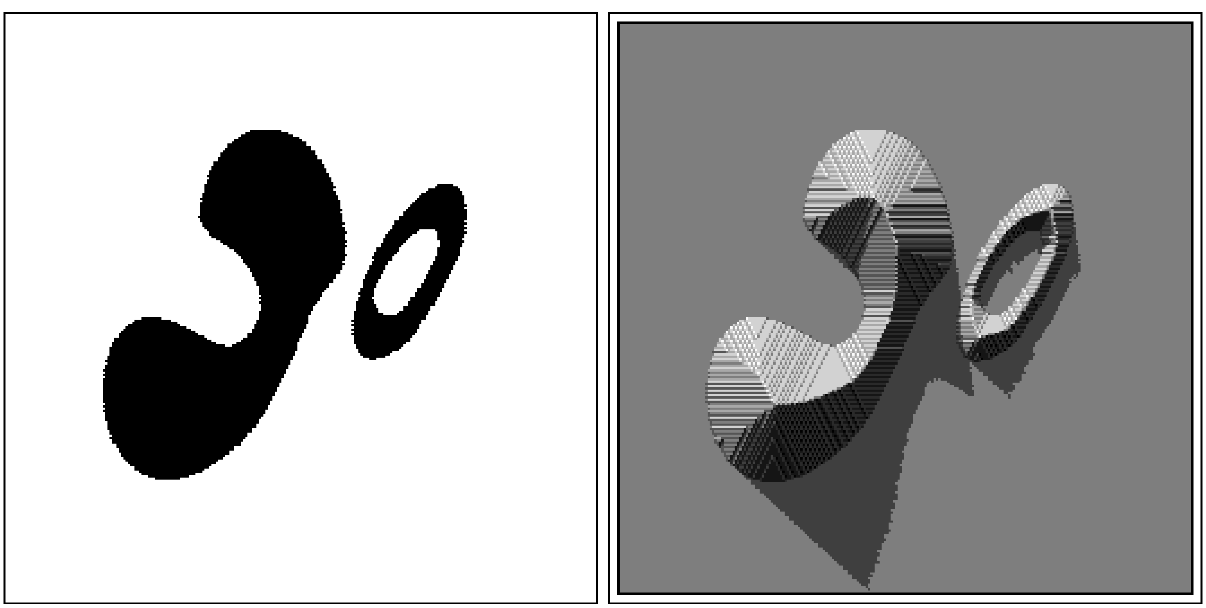
Parallel and sequential algorithms

In morphology (Serra, 1982, 1988), the most commonly used agorithms are the *parallel* ones (Vincent, 1990, 1991b). Indeed, they were historically the first to be introduced in the field and remain widely used on many systems. Their principle is to modify the value of the current pixel p of the input image (n-dimensional array of pixels) according to the values of the pixels in a given neighborhood of p. The new value of p is then written in an output image *different from the current one,* so that the order in which pixels are scanned has no influence on the result. Further scannings can then be performed, until a given criterion is fulfilled (e.g., a certain number of scannings is achieved, or stability is reached).

These algorithms are conceptually very easy and their implementation is straightforward. In addition, they are well suited to specialized hardware systems. However, they usually require a large number of complete image scannings, so that their interest on classical architectures—like workstations or personal computers—remains limited, due to prohibitive computation times. This is particularly true when complex transformations like watersheds or skeletons are considered.

For this reason, another family of algorithms was introduced in 1966: the so-called *sequential* or *recursive* algorithms (Rosenfeld *et al,* 1966; Laÿ, 1987). They aim at dramatically reducing the number of image scannings required to compute a given transformation. Like parallel ones, sequential algorithms do not require sophisticated data structures or scanning techniques. They differ from parallel algorithms in that they use well-defined scanning orders (usually video or anti-video) and that the value of the current pixel p, determined from the values of the pixels in its neighborhood, is written *in the image being processed.* So, the value of an already scanned pixel may have an influence on the value given to any pixel scanned later.

Although the contour-based methods described in the next two sections perform even better in this case, one of the transformations which is best suited to sequential methods is the *distance function.* This is a gray-level image $d(I)$ determined from a binary one I by associating with each feature pixel its distance to the background (see Fig. 1.a). Well-know sequential algorithms introduced in 1968 (Rosenfeld, 1968) allow us to determine distance functions in two image scanning (Vincent, 1990, 1991b). On Fig. 1.b, the distance function of 1.a has been artificially shaded and shadowed using a morphological dilation by a 3-D segment. This step was also implemented using a sequential algorithm detailed in (Vincent, 1989). Note that sequential distance function algorithms have been recently built into the morphology chip PIMM1 (Klein *et al.,* 1989).



(a) original image *(b) distance function with artificial shadowing*

Figure 1: Distance function of a binary shape.

Another example concerns the so-called *granulometry function* $g(I)$ of a binary picture I (Matheron, 1975). Given a family of convex homothetic structuring elements (B_n), this function associates with each pixel p the first integer $n \geq 0$ such that $\gamma_n(I)(p) = 0$, γ_n denoting the opening by B_n. When the B_n's are either hexagons or squares, the granulometry function can be efficiently determined via sequential algorithms. An example of a granulometry function by squares is shown in Fig. 2. Its sequential computation took only a few seconds on a Macintosh. In addition, this algorithm can be adapted to the determination of gray-scale dilations with segments, squares or hexagons.

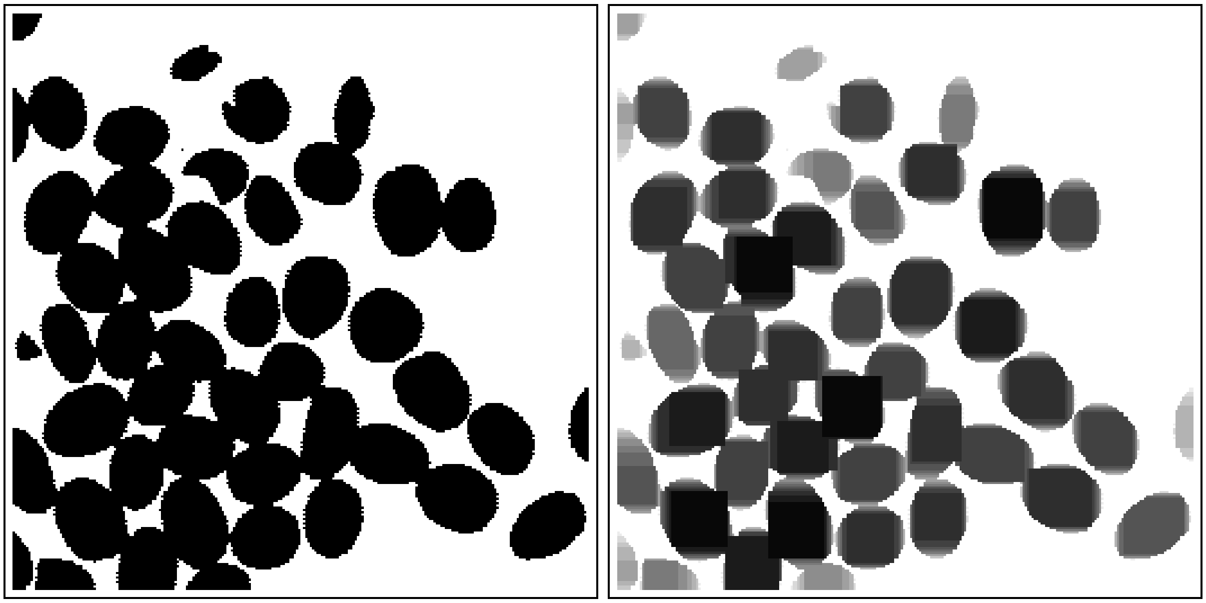


Figure 2: Binary image and corresponding granulometry function using squares as structuring elements.

Other operations whose implementation can be approached through recursive algorithms include reconstruction and other geodesic transformations, skeletons, prunings, etc. However, in all these cases, several image scannings are usually performed in which only very few pixels are modified.! Much more powerful algorithms can be proposed for computing such operations (Schmitt, 1989; Vincent, 1990, 1991b): their common denominator is to be based on contours, so that at each "step", only the pixels which may be modified are accounted for. The rest of the paper is devoted to a schematic presentation of these algorithms. For full details, see (Schmitt & Vincent, 1992).

Algorithms based on chains and loops

These techniques, introduced in 1989 by Schmitt for the hexagonal case, start from the remark that the boundary of an isotropic dilation $\delta(X)$ of a set X is a curve which is parallel to the boundary of X. Since most morphological operations can be described as combinations of elementary isotropic dilations, the principles adopted here are the following:

- Model the boundaries of the image under study as a set of loop data structures (Freeman chain codes). This is the contour tracking step.
- Propagate these structures in the image using *rewriting rules.*

In the propagation step, successive dilations of the loops are determined extremely quickly, since there is no need to look for information in the image itself once the loops have been extracted. The rewriting rules used for dilating hexagonal loops are shown in Fig. 3. After each dilation step, an *adjustment* step is inserted: it consists in writing the loops in the image, giving the corresponding pixels an appropriate value, and in cutting the unnecessary loop parts, thereby creating chains. These chains (nonclosed loops) are then manipulated exactly as loops.

Loops and chains algorithms are very efficient for most binary 2-D morphological operations, but perform especially well in the geodesic case (Lantuéjoul *et al.,* 1984). To compute such transformations as, e.g., *geodesic distance function, reconstruction* and *labelling,* one single scan of the feature pixels is needed—after the initial contour extraction. In addition, the present category of algorithms provides the only known technique to efficiently determine *propagation functions* in binary images (Schmitt, 1989; Maisonneuve & Schmitt, 1989). Recall that the propagation function associates with each pixel of a connected set X its geodesic distance to the farthest pixel of X (see Fig. 4).

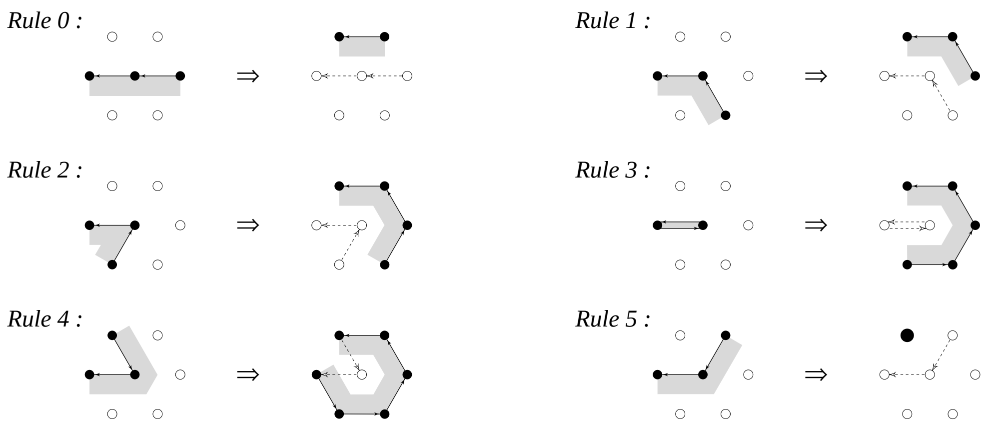


Figure 3: Rewriting rules allowing to dilate chains and loops.

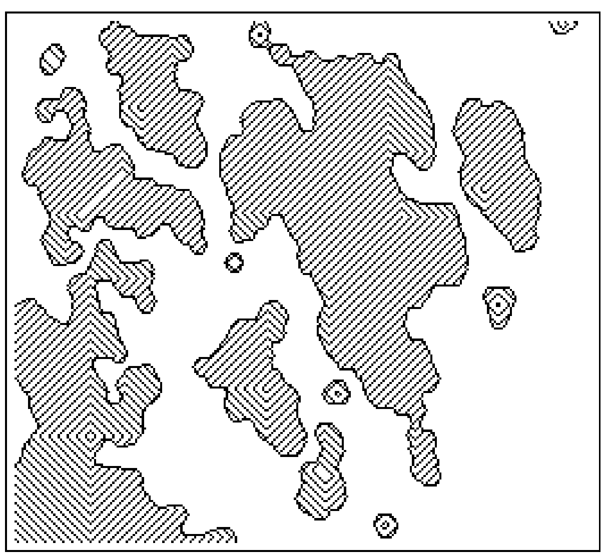


Figure 4: Level lines of the 4-connected propagation function of a binary image.

Euclidean distance functions and applications

Loops and chains techniques can also be adjusted in such a way that they allow the fast computation of *exact* Euclidean distance functions (Vincent, 1990, 1991d). The idea is to modify the chain and loop structure as well the rewriting rules of Fig. 3 in such a way that Euclidean distances are conveyed in the image by theses structures. Previous algorithms were of sequential type and only yielded more isotropic distances (Borgefors, 1986) or approximations of Euclidean distance functions (Danielsson, 1980). An example of exact Euclidean distance function is showed in Fig. 5. The same techniques are used to determine Euclidean skeletons by influence zones (Lantuéjoul, 1980), Delaunay triangulations and Gabriel graphs in arbitrary binary pictures, as illustrated by Figs. 6 and 7 (Vincent 1990, 1991d).

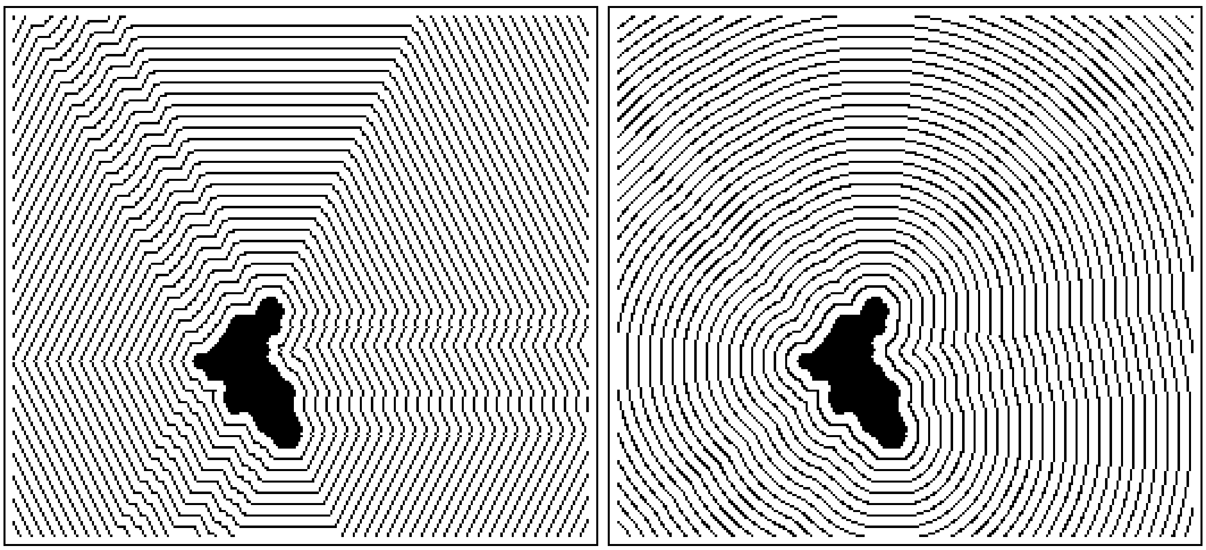


Figure 5: Comparison between hexagonal and Euclidean distance functions.

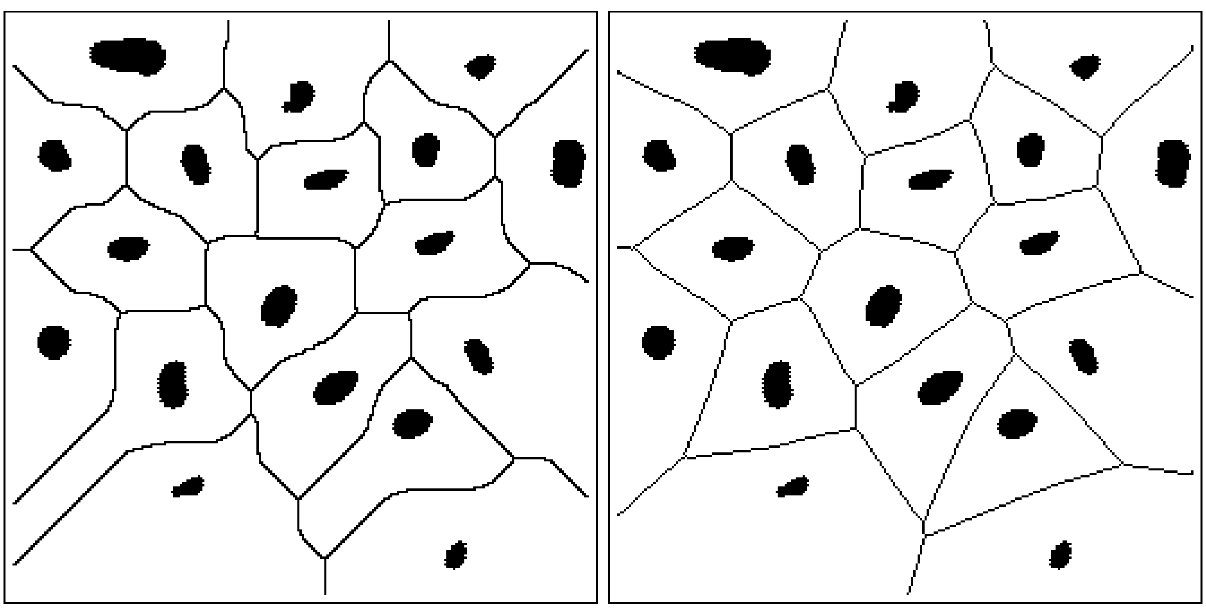


Figure 6: Chessboard versus Euclidean skeleton by influence zones (SKIZ).

Binary operations using arbitrary structuring elements

Other extensions of the present loop based methods include efficient algorithms for computing binary dilations, erosions, openings and closings with structuring elements of arbitrary size and shape (Vincent, 1990, 1991a). Here, chains and loops are no longer propagated in the image. Instead, we encode the

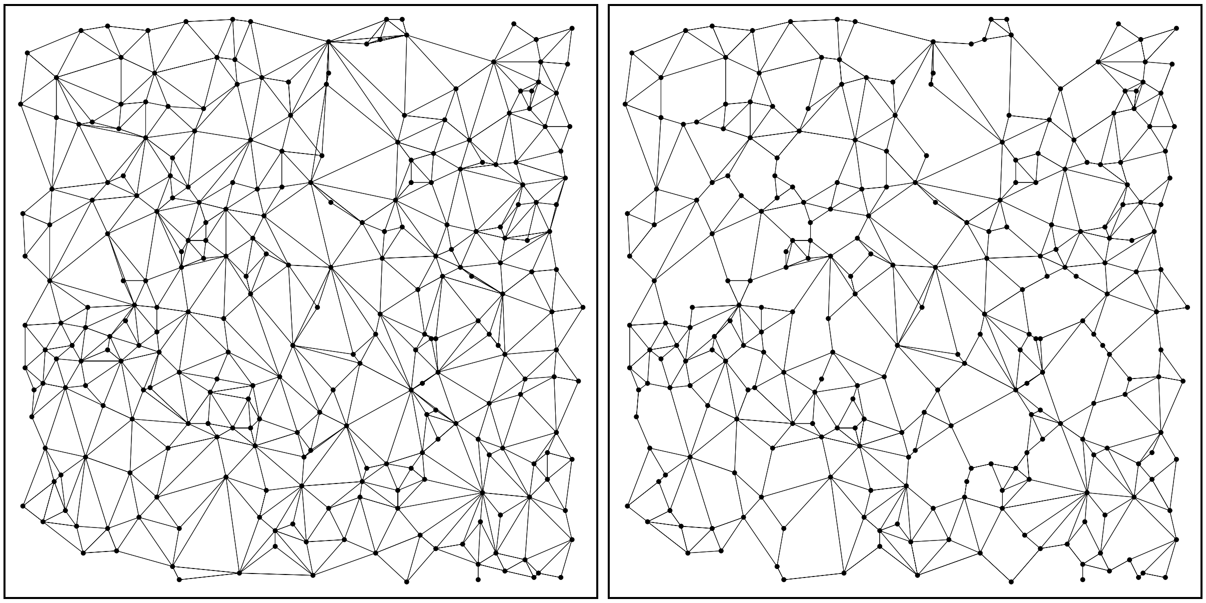


Figure 7: Delaunay Triangulation and Gabriel Graph in a digital binary image.

involved structuring element appropriately and propagate it along the loops representing the set (binary image) to be dilated or eroded. Combining an erosion and a dilation step allows us to determine openings and closings equally well, as illustrated by Fig. 8.

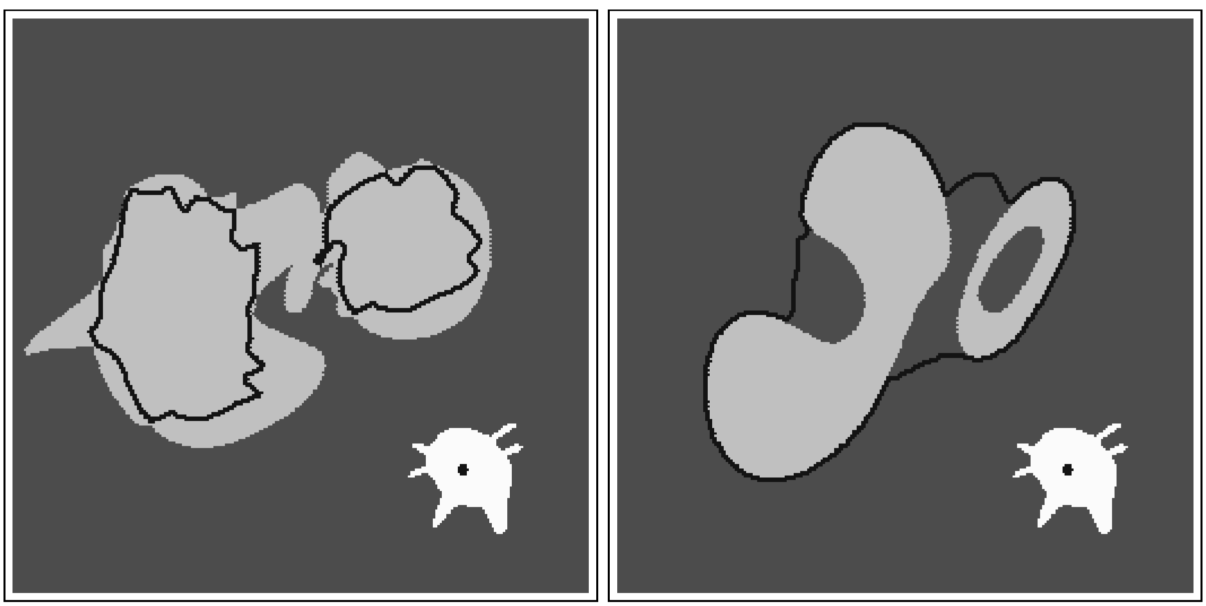


Figure 8: Binary opening and closing by an arbitrary (and weird!) structuring element.

FIFO ALGORITHMS

The second main family of contour-based algorithms which is of great interest in morphology is that of the *queue-based algorithms* (Van Vliet, 1988; Vincent, 1990). Their principle is to regard the image under study as a graph whose nodes are the pixels themselves and whose edges are provided by the used grid (4-, 6- or 8-connectivity usually). Starting from the objects boundaries, breadth-first scannings of this graph are performed, in which pixels belonging to the successive dilations are progressively reached. This process is enabled by a *First-In-First-Out (FIFO)* data structure which is a *queue of pixels*.

In such a structure, the pixels which are first put into it are those which can first be extracted: each pixel included in the queue is put on one side whereas each pixel being removed is taken from the other side (see Fig. 9). To perform breadth-first scanning of an image starting from a given set of pixels, the first step is to include them in the queue. Then, until stability is reached, the first pixel of the queue is extracted and its not yet considered neighbors are put on the queue themselves. Throughout the process, the value of a pixel indicates whether it has already been considered or not.

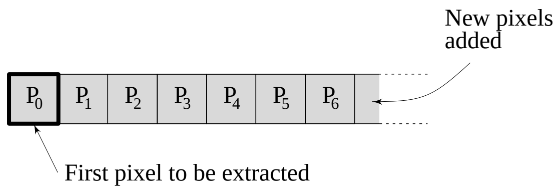


Figure 9: How a queue of pixels works.

Most classical binary operations that can be obtained via chains and loops algorithms are easily implemented using queues of pixels. In particular: Euclidean and geodesic dilations, erosions, openings and closings, distance functions, reconstruction, labelling, hole filling, etc. The computation costs are hardly higher than with chains and loops methods. However, the flexibility of FIFO techniques is tremendous: once an algorithm has been developed for a given image type, its adaptation to a different grid, to n-dimensional images or even to general graphs is straightforward. Indeed, as detailed in (Vincent 1990, 1991b), it suffices to modify the way the neighbors of a given pixel (resp., voxel, vertex, etc.) are generated. Furthermore, unlike every other family of algorithms, FIFO-based techniques allow us to efficiently compute such complex transformations as skeletons and watersheds. The rest of the paper is devoted to these transformations.

Skeletons

There has been a great number of methods proposed in literature for the determination of skeletons in digital pictures, some of the main ones are reviewed in (Vincent, 1990). For example, parallel methods make use of homotopic thinnings iterated until stability is reached. Other techniques work through the extraction of the *crest-lines* of the distance function. We show in (Vincent, 1991c) that these algorithms are either inaccurate, or slow, or suffer from a lack of flexibility.

On the other hand, FIFO structures together with the concept of *anchor point* form the basis of a particularly interesting skeleton algorithm, detailed in (Vincent, 1991c). The idea is to simulate the propagation of a grassfire inside the set X to be skeletonized, this fire being initialized at the boundary of X (see Fig.10). To realize that in practice, we do the following:

- Extraction of anchor points, i.e., points which surely belong to the final skeleton. Such points are easily determined as the local maxima of the distance function (Vincent, 1991c).
- Starting from the boundary pixels of X, breadth-first scanning of this set is performed. Every pixel p considered is removed if and only if:
  - it is not an anchor point,
  - removing it does not modify the local homotopy.

The last requirement assures that we end up with a *connected* skeleton, i.e., a skeleton which preserves the homotopy of the initial set. For these homotopy checkings, look-up tables are used, where every local configuration is given value 0 or 1 depending upon its homotopy status (Vincent, 1991c). This can be achieved in 4-, 6- or 8-connectivity equally well. For example, Fig. 11 shows an 8-connected skeleton of a binary image of lamellar eutectics, determined using the present technique.

Like all FIFO algorithms, the present one is particularly efficient: indeed, after the anchor points extraction and the queue initialization, only the feature pixels are considered in the fire propagation step. The resulting skeletons are also more accurate than with most other methods. The algorithm performs in the Euclidean case as well as in the geodesic one. Most interestingly, it allows us to calculate a whole range of different skeleton-like transformations, whose computation is hardly possible otherwise: this is simply achieved by using different sets of anchor points. For example, by taking as anchor points the *regional* maxima of the distance function instead of its local maxima, one gets an object referred to in literature as *minimal skeleton*. Similarly, an empty set of anchor points results in a "homotopic marking" of the set. Using as anchor points only the local maxima of elevation greater than n results in a smoother

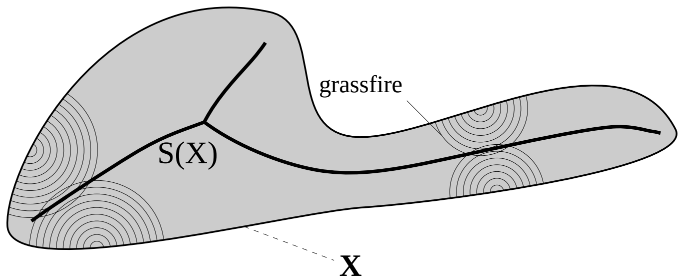


Figure 10: The skeleton can be viewed as the locus of the pixels where two firefronts stemming from the objects boundaries meet.

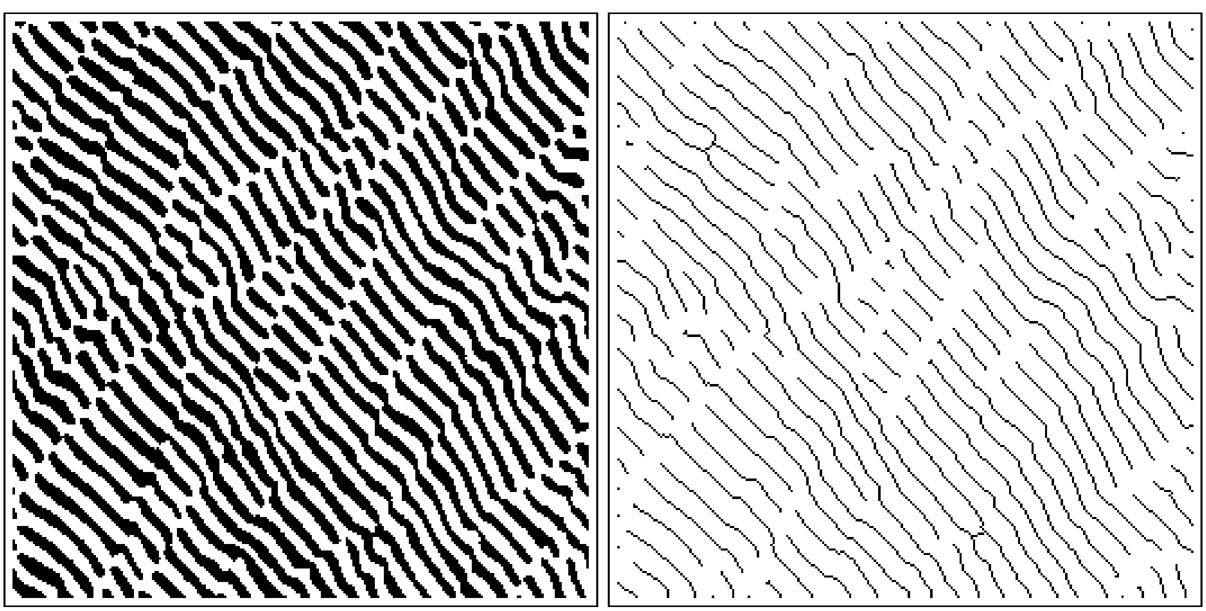
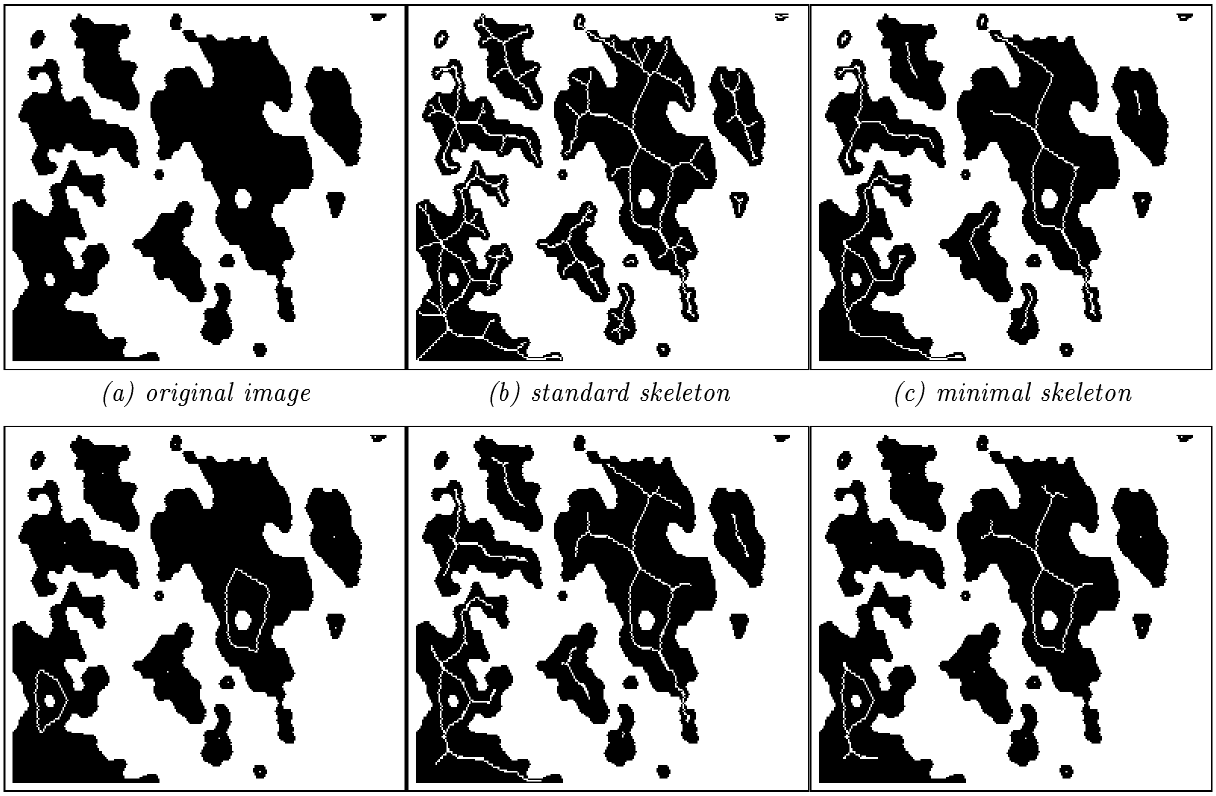


Figure 11: Binary image and its 8-connected skeleton obtained using the method described (followed by a 3 prunings iterations and 1 thinning step).

skeleton called *skeleton of order n* (Vincent, 1991c). Any of these skeletons can then be post-processed via prunings, themselves realized via FIFO algorithms. Fig. 12 shows a sample of these possibilities. To summarize, the present queue-based skeleton algorithm is particularly efficient, accurate and flexible.



(a) original image *(b) standard skeleton* *(c) minimal skeleton*

(d) homotopic marking *(e) pruned skeleton (16 iterations)* *(f) skeleton of order 16*

Figure 12: Various kinds of skeletons which may be efficiently determined using algorithms based on queues of pixels (Vincent, 1991c). This example was produced using hexagonal grid.

Watersheds

In the last decade, increasing attention has been put on the watershed trasformation as a tool for image segmentation (Digabel & Lantuéjoul, 1977; Beucher & Lantuéjoul, 1979; Beucher & Vincent, 1990). It is defined for grayscale images via the notion of a *catch basin:* let us regard the image under study as a topography (where the gray-level of a pixel stands for its altitude) and assume it is raining on it. A drop of water falling at a point p flows down along a steepest slope path until it is trapped in a minimum m of the relief. The set $C(m)$ of the pixels such that a drop falling on them eventually reaches m is called catch basin associated with minimum m. The set of the boundaries of the different catch basins of an image constitutes its watersheds. These notions are recalled in Fig. 13.

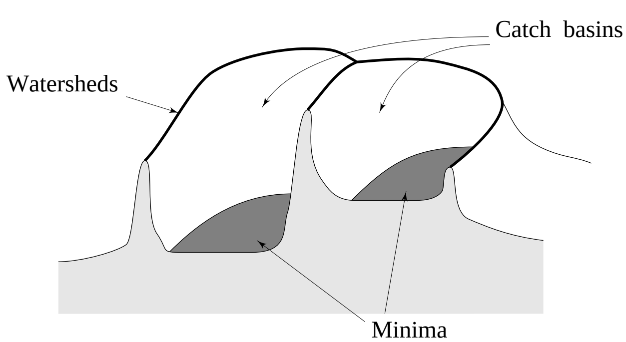


Figure 13: Minima, catch basins and watersheds.

Here again, various techniques have been proposed to determine watersheds in digital pictures. The major ones are reviewed in (Vincent, 1990; Vincent & Soille, 1991). The most interesting algorithmic idea, originally proposed by Beucher, consists in "inverting" the way of introducing the watershed notion: consider that the minima of the image—regarded here as a 3-D surface—have been pierced, and that this image is slowly immersed into a lake. The water progressively floods the different catch basins, and at some point, water originating from two different minima will merge, thereby connecting the corresponding catch basins. We prevent this by erecting dams at every place where this connection would otherwise occur. Once the surface is totally immersed, the set of dams thus built corresponds to the watersheds of the initial image.

This immersion and dam erection process can now be algorithmically simulated. The most efficient algorithm described in literature makes use of FIFO breadth-first scanning techniques for the actual flooding of the catch basins (Vincent & Soille, 1991). A labelling of the catch basins is also used, which automatically prevents the connection of two different basins. It has been shown that the results provided by this technique are more accurate that those of any other method. Just like almost all FIFO-algorithm, the present one extends to any grid and any dimension in a straightforward manner.

This algorithm dramatically reduces the computation times required for extracting watersheds. On conventional computers, previous approaches typically needed up to a couple of hours! The present one takes about 5 to 10 seconds on a Macintosh II, for a 256×256 image, thus opening the door to powerful segmentation methods on inexpensive software-based systems.

Let us conclude this section by an example of application of the present algorithm. We consider here Figure. 14.a: it is part of a series of successive images of the same scene, and the problem is to recover the motion of the camera. One of the approaches which has been considered to solve this motion estimation problem consists in decomposing the images into regions and to match these regions over successive time frames (Fuh & Maragos, 1989). Performing this decomposition by means of watershed techniques turns out to provide meaningful regions, which are then easily matched over time. As explained in detail in (Beucher & Vincent, 1990), the watershed tool is applied on the morphological gradient of the original image (Fig. 14.c). In fact, to avoid oversegmentation, the watersheds of the gradient are constrained by a *marker image.* The markers are connected components of pixels belonging to each of the regions to be extracted, and have been obtained here as "domes" and "basins" of the original image (see Fig. 14.b). The result of this constrained watershed transformation consists in the highest watershed edges of the gradient which are located between two markers (Fig. 14.d). The entire segmentation process takes approximately 5 seconds on a *Sun Sparc Station 2.*

Conclusion

In this paper, we dealt with four main categories of morphological algorithms: parallel, sequential, based on chains and loops, based on queues of pixels. Each of these techniques has its own advantages and drawbacks, so which one to choose for a particular purpose? First, if a specialized hardware with built-in elementary operations (dilation, erosion, thinning and thickening) is to be used, parallel algorithms still constitute the best choice for most operations. However, on a conventional architecture the game is much more open: sequential algorithms remain of interest for a number of specific transformations like distance functions, shadowing, grayscale geodesic reconstruction, granulometry functions and gray-level dilations with simple structuring elements. For almost all other operations, the contour-based algorithms described in the last two sections should be preferred:

- loops and chains techniques are best suited for any kind of binary operation, and are particularly efficient in the geodesic case (distance functions, propagation functions, labelling, reconstruction, etc).

- FIFO algorithms perform best for skeletons, SKIZ, watersheds and other complex transformations. They constitute one of the best choices when dealing with n-dimensional images.

Using the above guidelines, it is possible to design software systems where every operation is implemented efficiently. In particular, the recent advances in morphological algorithms reported in this paper enable the fast and accurate computation of transformations whose use was impossible before due to prohibitive computation times. Morphology is therefore now within the reach of virtually every individual. This makes it one of the leading images analysis methodologies of the 1990's.

Figure 14: Example of watershed segmentation: (a) original image, (b) marker image, (c) gradient image, (d) final segmentation obtained via watersheds of the gradient controlled by the markers.

Acknowledgements

This work was supported in part by *Dassault Electronique* and the *National Science Foundation* under Grant MIPS–86–58150, with matching funds from *DEC* and *Xerox.* Most of the examples of this paper have been produced using the software DIP Station™ on a Macintosh.

References

Borgefors G. Distance Transformations in Digital Images. Computer Vision, Graphics and Image Processing 1986; 34: 334–371.

Beucher S. & Lantuéjoul Ch. Use of Watersheds in Contour Detection. Proc. Int. Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation, Rennes, France, September 1979.

Beucher & L. Vincent. Introduction aux outils morphologiques de segmentation. In: Traitement d'images en microscopie à balayage et en microanalyse par sonde électronique. Paris: ANRT, 1990: F1–43.

Danielsson P.E., Euclidean Distance Mapping. Computer Graphics and Image Processing 1980; 14: 227–248.

Digabel H. & Lantuéjoul Ch. Iterative Algorithms. Proc. 2nd European Symposium on Quantitative Analysis of Microstructures in Material Science, Biology and Medicine, Caen, France, October 1977; Stuttgart (FRG): Riederer Verlag, 1978, J-L. Chermant Ed.: 85–99.

Fuh C-S. & Maragos P. Region Based Optical Flow Estimation. Proc. IEEE Computer Vision and Pattern Recognition Conference'89. San Diego, CA, June 1989: 130–135.

Klein J.C. & Peyrard R. PIMM1, an Image Processing ASIC Based on Mathematical Morphology. Proc. IEEE's ASIC Seminar and Exhibit. Rochester, NY, September 1989: 25–28.

Lantuéjoul Ch. Skeletonization in Quantitative Metallography. Issues of Digital Image Processing; R. M. Haralick & J-C. Simon eds. Groningen, The Netherlands: Sijthoff and Noordhoff, 1980.

Lantuéjoul Ch. & Maisonneuve F. Geodesic Methods in Image Analysis. Pattern Recognition 1984; 17: 117–187.

Laÿ B. Recursive Algorithms in Mathematical Morphology. Acta Stereol. 1987; 6/III. Proc. 7th Int. Congress For Stereology: 691–696.

Matheron G. Random Sets and Integral Geometry. New York: Wiley, 1975.

Maisonneuve F. & Schmitt M. An Efficient Algorithm to Compute the Hexagonal and Dodecagonal Propagation Function. Acta Stereol. 1989; 8/2: 515–520.

Rosenfeld A. & Pfaltz J.L. Sequential Operations in Digital Picture Processing. J. Assoc. Comp. Mach. 1966; 13 (4): 471–494.

Rosenfeld A. & Pfaltz J.L. Distance Functions on Digital Pictures. Pattern Recognition 1968; 1: 33–61.

Schmitt M. Des algorithmes morphologiques à l'intelligence artificielle. PhD Thesis. Paris: School of Mines, February 1989.

Schmitt M. & Vincent L. Morphology: Practical and Algorithmic Handbook for Image Analysis. London: Cambridge University Press, to appear in 1992.

Serra J. Image analysis and mathematical morphology. London: Academic Press, 1982.

Serra J. Image analysis and mathematical morphology, part II: theoretical advances. London: Academic Press, 1988.

Vincent L. Morphological shading and shadowing algorithm, Proc. PIXIM 89, Computer Graphics in Paris. Paris, September 1989; Hermès, Paris: 109–124.

Vincent L. Algorithmes morphologiques à base de files d'attente et de lacets. Extension aux graphes. PhD Thesis. Paris: School of Mines, May 1990.

Vincent L. Morphological Transformations of Binary Images With Arbitrary Structuring Elements. Signal Processing, January 1991; 22 (1): 3–23.

Vincent L. New Trends in Morphological Algorithms. Proc. SPIE/SPSE Nonlinear Image Processing II. San Jose, CA, February 1991; 1451: 158–170.

Vincent L. Efficient Computation of Various Types of Skeletons, Proc. SPIE Medical Imaging V. San Jose, CA, February 1991.

Vincent L. Exact Euclidean Distance Function by Chain Propagations. Proc. Computer Vision and Pattern Recognition'91. Maui, Hawaii, June 1991: 520–525.

Vincent L. & Soille P. Watersheds in Digital Spaces: an Efficient Algorithm Based on Immersion Simulations. IEEE Trans. Pattern Anal. Machine Intel. June 1991; 13 (6): 583–598.

Van Vliet L.J. & Verwer B.J.H. A Contour Processing Method for Fast Binary Neighborhood Operations. Pattern Recognition Letters 1988; 7: 27–36.