W-operator in its corresponding ROBDD. The uniqueness of the ROBDD representation allows a simple solution to the problem of checking the equivalence between morphological operators.

Currently, we are working on an implementation of a BDD-based morphological machine.

Acknowledgments

This work was partially supported by grant 1 D43 TW07015- 01 from the National Institutes of Health, USA.

References

[1] S. B. Akers. Binary Decision Diagrams. *IEEE Transactions on Computers*, C-27(6):509–516, June 1978.

[2] G. J. F. Banon and J. Barrera. Minimal Representations for Translation-Invariant Set Mappings by Mathematical Morphology. *SIAM J. Appl. Math.*, 51(6):1782–1798, December 1991.

[3] J. Barrera and G. P. Salas. Set Operations on Closed Intervals and Their Applications to the Automatic Programming of Morphological Machines. *Electronic Imaging*, 5(3):335–352, July 1996.

[4] G. Birkhoff. *Lattice Theory*. American Mathematical Society Colloquium Publications, Rhode Island, 1967.

[5] K. S. Brace, R. L. Rudell, and R. E. Bryant. Efficient Implementation of a BDD Package. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, pages 40–45. ACM/IEEE, 1990.

[6] R. E. Bryant. Graph-Based Algorithms for Boolean Function Manipulation. *IEEE Transactions on Computers*, C-35(8):677–691, August 1986.

[7] G. de Micheli. *Synthesis and Optimization of Digital Circuits*. McGraw-Hill Higher Education, 1994.

[8] H. J. A. M. Heijmans. *Morphological Image Operators*. Academic Press, Boston, 1994.

[9] C.Y. Lee. Representation of Switching Circuits by Binary-Decision Programs. *Bell Systems Technical Journal*, 38:985–999, July 1959.

[10] H. M. F. Madeira, J. Barrera, R. Hirata Jr., and N. S. T. Hirata. A New Paradigm for the Architecture of Morphological Machines: Binary Decision Diagrams. In *SIBGRAPI'99 - XII Brazilian Symposium of Computer Graphic and Image Processing*, pages 283–292. IEEE Computer Society, November 1999.

[11] H. M. F. Madeira and J. Barrera. Incremental Evaluation of BDD-Represented Set Operators. In *SIBGRAPI 2000 - XIII Brazilian Symposium of Computer Graphic and Image Processing*, pages 308–315. IEEE Computer Society, 2000.

[12] L. Robert and G. Malandain. Fast Binary Image Processing Using Binary Decision Diagrams. *Computer Vision and Image Understanding*, 72(1):1–9, October 1998.

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

IMAGE FILTERING USING MORPHOLOGICAL AMOEBAS

Romain Lerallut, Étienne Decencière and Fernand Meyer

Centre de Morphologie Mathématique, École des Mines de Paris

35 rue Saint-Honoré, 77305 Fontainebleau, France

lerallut@cmm.ensmp.fr

Abstract This paper presents morphological operators with non-fixed shape kernels, or amoebas, which take into account the image contour variations to adapt their shape. Experiments on grayscale and color images demonstrate that these novel filters outperform classical morphological operations with a fixed, space-invariant structuring element for noise reduction applications.

Keywords: Anisotropic filters, noise reduction, morphological filters, color filters

1. Introduction

Noise is possibly the most annoying problem in the field of image processing. There are two ways to work around it: either design particularly robust algorithms that can work in noisy environments, or try to eliminate the noise in a first step while losing as little relevant information as possible and consequently use a normally robust algorithm.

There are of course many algorithms that aim at reducing the amount of noise in images. Most are quite effective but also often remove thin elements such as canals or peninsulas. Even worse, they can displace the contours and thus create additional problems in a segmentation application.

In mathematical morphology we often couple one of these noise-reduction filters to a reconstruction filter that attempts to reconstruct only relevant information, such as contours, and not noise. However, a faithful reconstruction can be problematic when the contour itself is corrupted by noise. This can cause great problems in some applications which rely heavily on clean contour surfaces, such as 3D visualization, so a novel approach was proposed.

C. Ronse et al. (eds.), Mathematical Morphology: 40 Years On, 13–22.

2. Amoebas: dynamic structuring elements

Principle

Classic filter kernel. Formally at least, classic filters work on a fixed-size sliding window, be they morphological operators (erosion, dilation) or convolution filters, such as the diffusion by a Gaussian. If the shape of that window does not adapt itself to the content of the image (see figure 1), the results deteriorate. For instance, an isotropic Gaussian diffusion smooths the contours when its kernel steps over a strong gradient area.

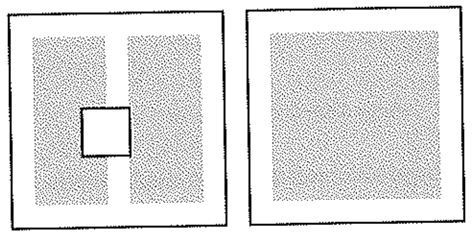


Figure 1 Closing of an image by a large structuring element. The structuring element does not adapt its shape and merges two distinct objects.

Amoeba filter kernel. Having made this observation, Perona and Malik [1] (and others after them) have developed anisotropic filters that inhibit diffusion through strong gradients. We were inspired by these examples to define morphological filters whose kernels adapt to the content of the image in order to keep a certain homogeneousness inside each structuring element (see figure 2). The coupling performed between the geometric distance between pixels and the distance between their values has similarities with the work of Tomasi and Manduchi described in [5].

The interest of this approach, compared to the analytical one pioneered by Perona and Malik is that it does not depart greatly from what we use in mathematical morphology, and therefore most of our algorithms can be made to use amoebas with little additional work. Most of the underlying theoretical groundwork for the morphological approach has been described by Jean Serra in his study [2] of structuring functions, although until now it has seen little practical use.

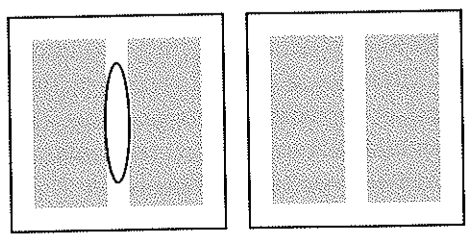


Figure 2 Closing of an image by an amoeba. The amoeba does not cross the contour and as such preserves even the small canals.

The shape of the amoeba must be computed for each pixel around which it is centered. Figure 3 shows the shape of an amoeba depending on the position of its center. Note that in flat areas such as the center of the disc, or the

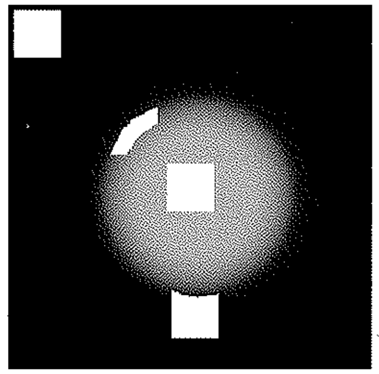


Figure 3 Shape of an amoeba at various positions on an image.

background, the amoeba is maximally stretched, while it is reluctant to cross contour lines.

When an amoeba has been defined, most morphological operators and many other types of filters can be used on it: median, mean, rank filters, erosion, dilation, opening, closing, even more complex algorithms such as reconstruction filters, levelings, floodings, etc.

Construction

Amoeba distance. In general, a filtering kernel of radius r is formally defined on a square (or a hexagon) of that radius, that is to say on the ball of radius r relative to the norm associated to the chosen connectivity. We will keep this definition changing only the norm, using one that takes into account the gradient of the image.

DEFINITION 1 *Let d_{pixel} be a distance defined on the* values *of the image, for example a difference of gray-value, or a color distance.*

Let $\sigma = (x = x_0, x_1, \ldots, x_n = y)$ a path between points x and y. Let λ be a real positive number. The length of the path σ is defined as

$$L(\sigma) = \sum_{i=0}^{n} 1 + \lambda . d_{pixel}(x_i, x_{i+1})$$

The "amoeba distance" with parameter λ is thus defined as:

$$\begin{cases} d_\lambda(x,x) &= 0 \\ d_\lambda(x,y) &= \min_\sigma L(\sigma) \end{cases}$$

It it important to realize that d_{pixel} has no geometrical aspect, it is a distance computed only on the values of the pixels of the image. Furthermore, if n is the number of pixels of a path σ, then $L(\sigma) \geq n$ (since $\lambda \geq 0$), which bounds the maximal extension of the amoeba.
This distance also offers an interesting inclusion property:

PROPERTY 1 *At a radius r given the family of the balls $\mathcal{B}_{\lambda,r}$ relative to the distance d_λ is decreasing (for the inclusion),*

$$\begin{aligned} 0 \leq \lambda_1 \leq \lambda_2 &\Rightarrow \forall(x,y), d_{\lambda_1}(x,y) \leq d_{\lambda_2}(x,y) \\ &\Rightarrow \forall r \in \mathbf{R}^+, \mathcal{B}_{\lambda_1,r} \supset \mathcal{B}_{\lambda_2,r} \end{aligned}$$

Which may be useful when building hierarchies of filters, such as a family of alternate sequential filters with strong gradient-preserving properties.

The pilot image. We have found that the noise in the image can often distort the shape of the amoeba. As such, we often compute the *shape* of the amoeba on another image. Once the shape is computed, the values are sampled on the *original* image and processed by the filter (mean, median, max, min, ...). Usually, the other image is the result of a strong noise removal filtering of the original image that dampens the noise while preserving as much as possible the larger contours. A large Gaussian works fairly well, and can be applied very quickly with advanced algorithms, however we will see below that iterating amoeba filters yields even better results.

3. Amoebas in practice

Adjunction

Erosions and dilations can easily be defined on amoebas. However it is necessary to use *adjoint* erosions and dilations when using them to define openings and closings:

$$\begin{aligned} \delta(X) &= \bigcup_{x\in X} B_{\lambda,r}(x) \\ \epsilon(X) &= \{x / B_{\lambda,r}(x) \subset X\} \end{aligned}$$

These two operations are at the same time adjoint and relatively easy to compute, contrary to the symmetrical ones that use the transposition, which is not easy to compute for amoebas. See [2] for a discussion of the various forms of adjunction and transposition of structuring functions.

Algorithms

The algorithms used for the erosion and dilation are quite similar to those used with regular structuring elements, with the exception of the step of computing the shape of the amoeba.

***Erosion (gray-level)*:**
for each pixel x:
  compute the shape of the amoeba centered on x
  compute the minimum$M$ of the pixels in the amoeba
  set the pixel of the output image at position x to value M

Dilation (gray-level):
for each pixel x:
  compute the shape of the amoeba centered on x
  for each pixel y of the amoeba:
    value(y)=max(value(y),value(x))

The opening using these algorithms can be seen as the gray-level extension of the classic binary algorithm of first taking the centers of the circles that fit inside the shape (erosion), and then returning the union of all those circles (dilation).

Complexity

The theoretical complexity of a simple amoeba-based filter (erosion, dilation, mean, median) can be asymptotically approximated by:

$$T(n,k,op) = O\left[n * \left(op(k^d) + amoeba(k,d)\right)\right]$$

Where n is the number of pixels in the image, d is the dimensionality of the image (usually 2 or 3), k is the maximum radius of the amoeba, $op(k^d)$ is the cost of the operation and $amoeba(k,d)$ is the cost of computing the shape of the amoeba for a given pixel.

The shape of the amoebas is computed by a common region-growing implementation using a priority queue. Depending on the priority queue used, the complexity of this operation is in slightly more than $O(k^d)$ (see [3] and [4] for advanced queueing data structures).

Therefore, for erosion, dilation or mean as operators, we have a complexity of a little more than $O(n * k^d)$ which is the complexity of a filter on a fixed-shape kernel. It has indeed been verified in practice that, while being quite slower than with fixed-shape kernels (especially optimized ones), filters using amoebas tend to follow rather well the predicted complexity, and do not explode (tests have been performed on 3D images, size 512x512x100, with amoebas with sizes up to 21x21x21).

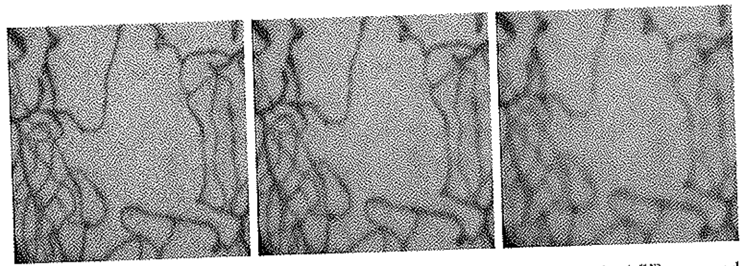
4. Results

Alternate sequential filters

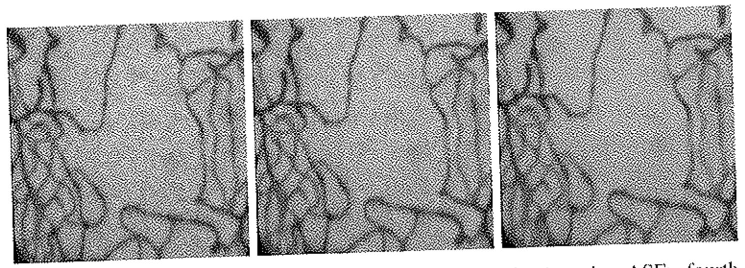
The images of figure 4 compare the differences between alternate sequential filters built on classic fixed shape kernels and ASFs on amoebas in the filtering of the image of a retina.

Median and mean

In the context of image enhancement, we have found that a simple mean or median coupled with an amoeba forms a very powerful noise-reduction filter.



(a) Original (b) Normal ASF: first pass (c) Normal ASF: second pass



(d) Amoeba ASF: first pass (e) Amoeba ASF: second pass (f) Amoeba ASF: fourth pass

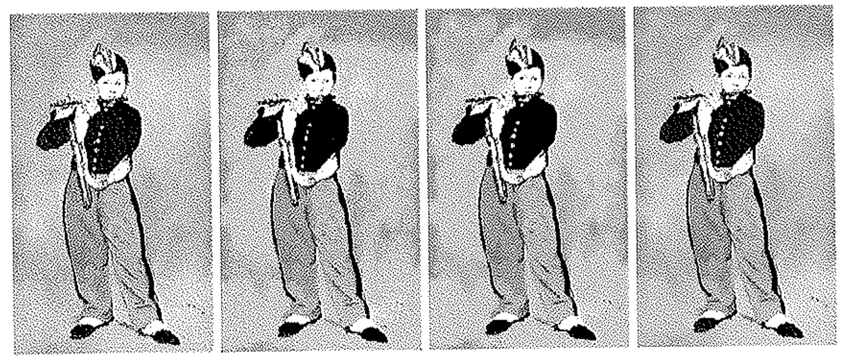
Figure 4. Alternate sequential filters on classic kernels and on amoebas. The amoeba preserves extremely well the blood vessels while strongly flattening the other areas.

The images in figure 5 show median and the mean computed on amoebas compared to those built on regular square kernels. The pilot image that drives the shape of the amoeba is the result of a standard Gaussian filter of size 3 on the original image, and the distance d_{pixel} is the absolute difference of gray-levels.

For the filters using amoebas, the median filter preserves well the contour, but the mean filter gives a more "aesthetically pleasing" image. In either case, the results are clearly superior to filterings by fixed-shape kernels, as seen in the figure 5.

Mean and median for color images

In the case of color images, the mean is replaced by the mean on each color component of the RGB color space. For the "median", the point closest to the barycenter is chosen. Other distances or colorspaces can be used, such as increasing the importance of the chrominance information with respect to



(a) Original (b) Usual median (c) Amoeba median (d) Amoeba mean

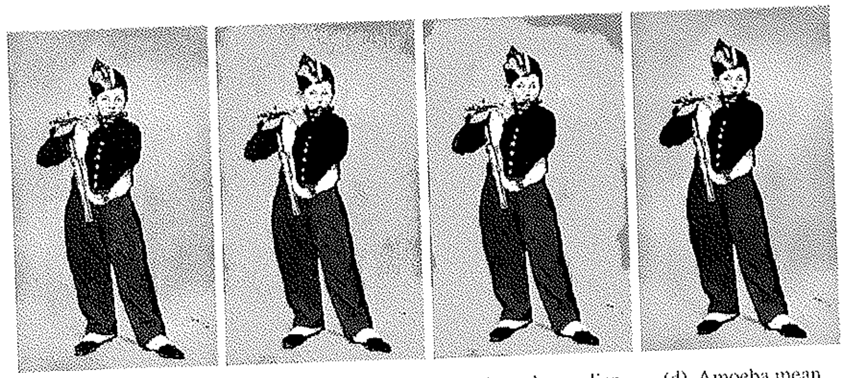
Figure 5. Results of a "classic" median filtering and two amoeba-based filterings: a median and a mean on Edouard Manet's painting "Le fifre".

luminance, or the other way around, depending on the application, the type of noise and the quality of the color information.

Iteration

The quality of the filtering strongly depends on the image that determines the shape of the amoeba. The previous examples have used the original image filtered by a Gaussian, but this does not always yield good results (also see [6]).

It is frequent indeed that a small detail of the image be excessively smoothed in the pilot image, and thus disappears completely in the result image. On the other hand, noisy pixels may be left untouched if the pilot image does not eliminate them. A possible solution is to somewhat iterate the process, using the first output image not as an input for filtering, as it would commonly be done, but as a new *pilot* image instead.



(a) Original (b) Usual median (c) Amoeba median (d) Amoeba mean

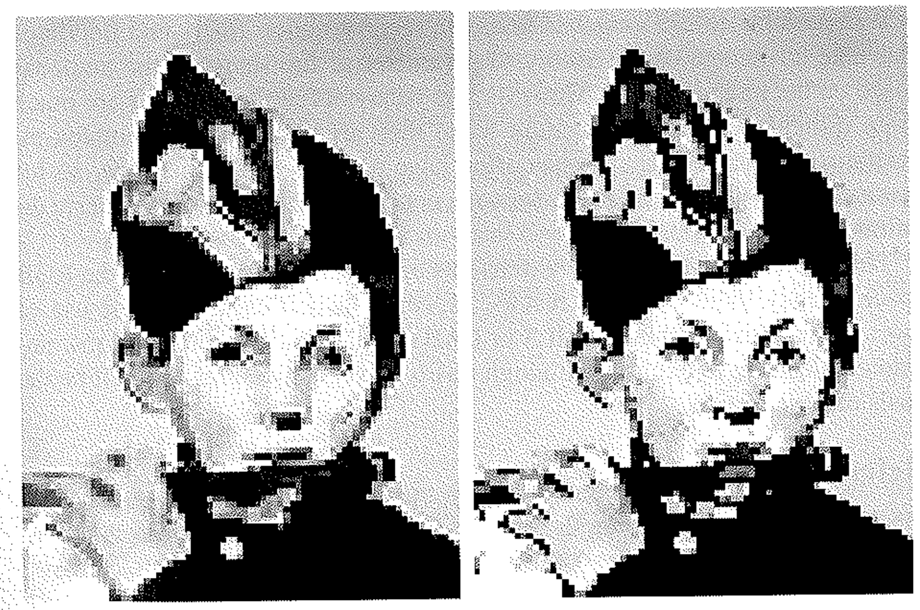
Figure 6. Color images: results of a "classic" median filtering, and two amoeba-based filterings: a median and a mean. As a simple extension of the grayscale approach, each channel of the pilot image has been independently smoothed by a Gaussian of size 3.

There are two steps at each iteration: the first one follows the scheme described earlier, using the Gaussian-filtered original image as a pilot, with aggressive parameters, and outputs a well-smoothed image in flat areas while preserving as much as possible the most important contours. The second step takes the original image as input and the filtered image as a pilot, with less destructive parameters, and preserves even more the finer details, while removing a lot of the noise.

In practice, we have found that performing those two steps only once is enough to reduce the noise dramatically (see figure 7), although further iterations may be required, depending on the image and the noise.

This method is also very useful for color images, since the amoeba-based pilot image provides better color coupling through the use of an appropriate color distance than simply merging the results of a Gaussian filtering of each channel independently.

(a) Original (b) Pilot image: Gaussian filter (c) Pilot image: amoeba mean filter



(d) Result image: amoeba mean with Gaussian pilot (e) Result image: amoeba mean with amoeba pilot

Figure 7. Comparison between two pilot images: a Gaussian one, and one based on a strong amoeba-based filtering. With the amoeba pilot image the hand is better preserved, and the eyebrows do not begin to merge with the eyes, contrary to the Gaussian-based pilot image. Having both less noise and stronger contours in the pilot image also enables the use of smaller values on the lambda parameter so that the amoeba will stretch more in the flatter zones, and thus have a stronger smoothing effect in those zones, while preserving the position and dynamics of the contours

5. Conclusion and future work

We have presented here a new type of structuring element that can be used in many morphological algorithms. By taking advantage of outside information, filters built upon those structuring elements can be made more robust on noisy images and in general behave in a "more sensible" way than those based on fixed-shape structuring elements. In addition, morphological amoebas are very adaptable and can be used on color images as well as monospectral ones and, like most morphological tools, they can be used on images of any dimension (2D, 3D, ...). Depending on the application, alternate sequential filters are very effective when looking for very flat zones, whereas median and mean filters output smoother images that may be more pleasing to the eye but could be harder to segment.

Work is currently in progress to integrate the filtered pilot image directly in the basic formulation, instead of having it as a preprocessing step, with the various drawbacks studied in [6].

It is possible to use amoebas to create reconstruction filters and floodings that take advantage of the ability to parameterize the shape of the amoebas based on the image content. However, the behaviors of the amoebas are much more difficult to take into account when they are used in such complex algorithms. In particular, amoebas often have a radius larger than one, so for instance the identification made between conditional dilation and geodesic dilation is no longer valid.

The results show that simple extensions of the scalar algorithms to the RGB space already yield excellent results, especially when iterating. The use of more "perceptual" distances (HLS or LAB) would probably prevent most unwanted blending of features, although this is as yet conjectural and will be the basis of further work.

References

[1] Perona, P. and Malik, J., "Scale-space and edge detection using anisotropic diffusion", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, July 1990

[2] Serra, J. *et al*, "Mathematical morphology for boolean lattices", *Image analysis and mathematical morphology - Volume 2: Theoretical advances*, Chapter 2, pp 37-46, Academic Press, 1988

[3] Cherkassky, B. V. and Goldberg, A V., "Heap-on-top priority queues", TR 96-042, NEC Research Institute, Princeton, NJ, 1996

[4] Brodnik, A. *et al.*, "Worst case constant time priority queue", Symposium on Discrete Algorithms, 2001, pp 523-528

[5] Tomasi, C. and Manduchi, R.,"Bilateral Filtering for Gray and Color Images", Proceedings of IEEE International Conference on Computer Vision, Bombay, India, 1998

[6] Catté, F. *et al.*, "Image selective smoothing and edge detection by nonlinear diffusion, SIAM J. Numerical Analysis, Vol. 29 No. 1, pp 182-193, February 1992

NUMERICAL RESIDUES

Serge Beucher

Centre de Morphologie Mathématique, Ecole des Mines de Paris
35, rue Saint Honoré,77300 Fontainebleau, France
Serge.Beucher@ensmp.fr

Abstract

Binary morphological transformations based on the residues (ultimate erosion, skeleton by openings, etc.) are extended to functions by means of the transformation definition and of its associated function based on the analysis of the residue evolution in every point of the image. This definition allows to build not only the transformed image itself but also its associated function, indicating the value of the residue index for which this evolution is the most important. These definitions are totally compatible with the existing definitions for sets. Moreover, they have the advantage of supplying effective tools for shape analysis on one hand and, on the other hand , of allowing the definition of new residual transforms together with their associated functions. Two of these numerical residues will be introduced, called respectively ultimate opening and quasi-distance and, through some applications, the interest and efficiency of these operators will be illustrated.

1. Introduction

In binary morphology there are some operators based on the detection of residues of parametric transformations. Among these operators, the ultimate erosion or the skeleton by maximal balls can be quoted. They can more or less easily be extended to greytone images. These extensions are however of little use because it is difficult to exploit them. This paper explains the reasons of this difficulty and proposes a means to obtain interesting information from these transformations. It also introduces new residual transformations and illustrates their use in applications.

2. Binary residues: reminder of their definition

Only operators corresponding to the residues of two primitive transforms will be addressed here. A residual operator θ on a set X is defined by means

C. Ronse et al. (eds.), Mathematical Morphology: 40 Years On, 23–32.