

MMach: a mathematical morphology toolbox for the KHOROS system

Junior Barrera

Universidade de São Paulo
Instituto de Matemática e Estatística
Rua do Matão, 1010
05508-900 São Paulo, SP
Brazil
E-mail: jb@ime.usp.br

Gerald Jean Francis Banon

Instituto Nacional de Pesquisas Espaciais
Divisão de Processamento de Imagens
Caixa Postal 515
12201-970 São José dos Campos, SP
Brazil

Roberto de Alencar Lotufo

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Caixa Postal 6101
13081 Campinas, SP
Brazil

Roberta Hirata, Jr.

Universidade de São Paulo
Instituto de Matemática e Estatística
Rua do Matão, 1010
05508-900 São Paulo, SP
Brazil

Abstract. *Mathematical morphology is a general theory that studies the decomposition of operators between complete lattices in terms of some families of simple operators: dilations, erosions, anti-dilations, and anti-erosions. Nowadays, this theory is largely used in image processing and computer vision to extract information from images. The KHOROS system is an open and general environment for image processing and visualization that has become very popular. One of the main characteristics of KHOROS is its flexibility, since it runs on standard machines, supports several standard data formats, uses a visual programming language, and has tools to help the users to build in and install their own programs. A set of new programs can be organized as a subsystem called a toolbox. We present MMach, a fast and comprehensive mathematical morphology toolbox for the KHOROS system dealing with 1-D and 2-D grayscale and binary images. Each program that is applicable to grayscale and binary images has specialized algorithms for each of these data types, and these algorithms are chosen automatically according to the input data. Several examples illustrate applications of the toolbox in image analysis. © 1998 SPIE and IS&T. [S1017-9909(98)01701-2]*

Paper 96-061 received November 4, 1996; revision received June 13, 1997; accepted for publication August 1, 1997.
1017-9909/98/\$10.00 © 1998 SPIE and IS&T.

1 Introduction

Mathematical morphology is a solid algebraic theory to study operators (i.e., transformations) between complete lattices,^{1,2} and is an extremely powerful tool to extract image information.^{3,4}

From a theoretical point of view, mathematical morphology studies the decomposition of operators between complete lattices in terms of some families of simple operators: dilations, erosions, anti-dilations, and anti-erosions. These operators are called the *elementary operators of mathematical morphology*.

In mathematical morphology, operators are built by combining the elementary operators through the union, intersection, and composition operations. Once an operator is built, it can also be used as a primitive to build other operators and so on. This decomposition procedure can be described by a formal language, and a particular implementation of this language is called a *morphological machine* (MMach). The programs for an MMach are also called *morphological operators*.

From a practical point of view, an MMach is a tool to extract image information. Usually a goal is broken heuristically in subgoals that are achieved by primitive operators.

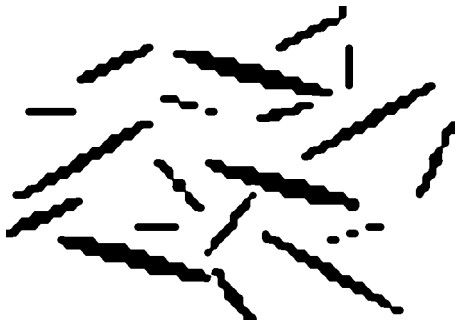


Fig. 1 Ellipses of several sizes.

The right composition of these primitives gives the operator that achieves the desired goal. For example, to extract long ellipses from images like the one in Figure 1, we could transform the ellipses into line segments (Figure 2), eliminate iteratively the end points of these segments until the segments representing the small ellipses disappear (Figure 3), and finally, recover from the original image the ellipses marked by the remaining fragments of the segments (Figure 4).

Thus a good system to perform mathematical morphology applications must have two main characteristics: fast algorithms for the elementary operators and a suitable interface to prototype new operators.

The KHOROS system is a portable environment for image processing and visualization that has become very popular. It runs on several UNIX based architectures, has a visual programming language for user interface, and provides tools to build and install new programs.

As the original set of programs for mathematical morphology in KHOROS is not satisfactory, we decided to implement a toolbox dedicated to image processing by mathematical morphology. This toolbox, called MMach, was first freely distributed in 1993. Now in its fourth version, it has hundreds of users and is still freely distributed at <http://www.dca.fee.unicamp.br/projects/khoros/mmach/tutor/mmach.html>.

An important characteristic of MMach is that its implementation follows exactly a formal specification of the operators and operations. In this paper, we give the formal specification of MMach as well as several illustrative application examples and the main characteristics of its architecture.

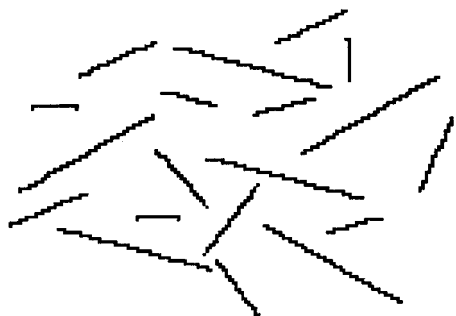


Fig. 2 Medial axis of the ellipses.



Fig. 3 Markers for the long ellipses.

We believe that in addition to reading this text through once, the readers may want to consult the mathematical definitions. For this purpose, Table 1 is useful, since having the name of the operator it is straightforward to get the number of the corresponding mathematical expression.

Following this introduction, Section 2 gives the formal specification of the implemented toolbox. Section 3 presents several application examples. Section 4 presents the main characteristics of the KHOROS system. Section 5 gives some aspects of the toolbox architecture and discusses the main algorithms implemented. Section 6 presents a comparison of MMach with other freely available software for mathematical morphology. Section 7 summarizes the main contributions of this paper and gives some directions for future works.

2 Morphological Image Processing

It has been shown^{5,6} that any operator between complete lattices can be described in terms of the four classes of elementary operators of mathematical morphology.

This result means that by combining the elementary operators implemented in our toolbox, one can create, in theory, any image processing algorithm. In practice, many useful operators can be implemented efficiently using decomposition in terms of elementary operators. However, for some useful operators this kind of implementation implies bad performance.

Therefore, given priority for developing an efficient computational tool, we decided to implement hierarchically most of the operators, but keep special algorithms for some particular operators. So the kernel of the toolbox contains a reduced set of chosen operators besides the elementary operators.

At the beginning, we decided to organize the MMach menu hierarchically, based on the number of elementary operators used in their decomposition. Thus, taking an increasing complexity order, we defined the following families of operations and operators: basic image operations and

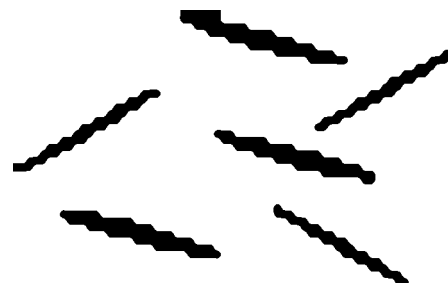


Fig. 4 Long ellipses.

Table 1 Toolbox content.

Toolbox Operators: MMach-1.4			
Category	Operator	Executable	Expression
Image Creation	Frame	vframe	1
Relations	Equal	vequal	3
	Less or Eq	vless	2
Operations	Addition	vaddm	10
	Ext.Intersection	vextintersec	6
	Ext.Union	vextunion	7
	Intersection	vintersec	4
	Negation	vneg	8
	Subtraction	vsubm	9
	Sym-Differ.	vsymdif	12
	Toggle	vtoggle	11
	Union	vunion	5
Structural Element	3×3 Str.El.	vstrflat	
	Cylinder	vcylinder	22
	Disk-City Block	vdiskstr	20, 18
	Disk-Euclidian	vdiskstr	20, 17
	Disk-Chess Board	vdiskstr	20, 19
	Rotate-SE	vrotstr	27
	Semi-sphere	vsphere	21
	Viff to Str.El.	viff2str	
Dilation and Erosion	Dilation	vdil	31
	Dilation by Segment	vdilseg	39
	Dist. Function	vdist	43
	Erosion	vero	32
	Erosion by Segment	veroseg	40
	N Dilation.	vndil	13, 31
	N Erosion	vnero	13, 32
Geometric Manipulation	Expansion by Dilation or Erosion	vexpand	48, 49
	Shrink by Dilation or Erosion	vshrink	50, 51
Morphological Filters	Center Filter	vcentfilt	13, 63
	Closing	vclose	53
	Closing by Segment	vcloseseg	57
	co-ASF	vcoASF	59, 62
	coc-ASF	vcocASF	61, 62
	oc-ASF	vocASF	58, 62
	oco-ASF	vocoASF	60, 62
	Opening	vopen	52
	Opening by Segment	vopenseg	56
	N-Closing	vnclose	55
	N-Opening	vnopen	54
Connected Filters	Cond. Dilation	vconddil	64
	Cond. Erosion	vcondero	65
	Closing of Holes	vclohole	68
	Inf-Reconstruction	vopenrec	66
	Labeling	vlabelm	69
	Minima Imposition	vminimpos	70
	N Cond. Dilation	vnconddil	13, 64
	N Cond. Erosion	vncondero	13, 65
	Regional Max	vrlocmax	71
	Regional Min	vlocmin	72
	Sup-Reconstruction	vclorec	67

Table 1 Continued.

Toolbox Operators: MMach-1.4			
Category	Operator	Executable	Expression
Residues	Close Reconst. Top Hat	vcloserecth	77
	Last Erosion	vlastero	80
	Morph. Close Top Hat	vcloseth	75
	Morph. Gradient	vmorphgrad	73
	Morph. Open Top Hat	vopenth	74
	Morph. Skel.	vskel	79
	N-Cond. Bisector	vncondbisec	81
	Objects on Frame Off	vedgeoff	78
	Opening Reconst. Top Hat	vopenrecth	76
Sup-generating and Inf-generating	Adaptive Threshold	vthreshm	82
	Inf-Generating	vinfgen	87
	Inf-Canonical	vicanon	92
	Sup-Generating	vsupgen	86
	Sup-Canonical	vscanon	91
Thinning and Thickening	Cond.Exoskel by Thick	vcondskthick	104
	Cond.Skel by Thin	vcondskthin	103
	Cond.Thick	vcondthick	96
	Cond.Thin	vcondthin	95
	Exoskel by Thick	vkthick	100
	N-Cond.Thick	vncondthick	102
	N-Cond.Thin	vncondthin	101
	N-Smoothing Segm.	vsmoothseg	106
	N-Thick	vnthick	98
	N-Thin	vnthin	97
	Refined Skel by Thin	vskelthinP	108
	Skiz	vskiz	105
	Skel by Thin	vskthin	99
	Thinning	vthin	93
Thickening	vthick	94	
Watershed	vwatersh	107	

operators, and first, second, and third level image operators. Although this organization is a clear criteria, it is not intuitive for users without a deep knowledge of the operator's decomposition. Therefore, from MMach version 1.4, we have reorganized the toolbox menu in the following families of operations and operators: relations, operations, structural elements, dilations and erosions, geometric manipulation, morphological filters, connected filters, residues, sup-generating and inf-generating, thinning, and thickening.

2.1 Image Representation

Let \mathbb{Z} be the set of integers, E be a rectangle of \mathbb{Z}^2 , and K be an interval $[0, k]$ of \mathbb{Z} , with $k > 0$. The collection of functions from E to K will represent the gray-scale images of interest. We denote such collection by K^E and its generic elements by f, g, f_1, f_2, f_i and f_n . When it is certain that we are dealing just with functions, a constant function in K^E will be denoted by the element of K which character-

izes it, for example, 0, 1, k . When $K = \{0, k\}$, the set K^E will represent the binary images. A binary image f may be represented equivalently by the subset F of E such that $x \in F \Leftrightarrow f(x) = k$.

Figure 5 presents a binary image. Figures 6 and 7 present, respectively, a gray-scale image and the graph of its representation as a function.

Let ∂E denote the limit points of the rectangle E . The function in $\{0, k\}^E$ given by

$$f(x) = \begin{cases} k & \text{if } x \in \partial E \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

is called the frame of E .

2.2 Relations

One of the most fundamental notions in mathematical morphology is the less or equal relation that induces the notion

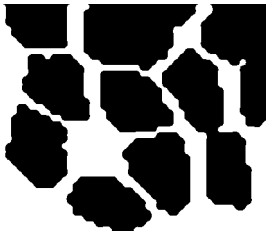


Fig. 5 Binary image.

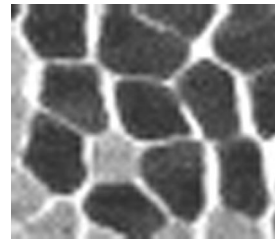


Fig. 6 Gray-scale image.

of partial ordering in a set of objects. In an MMach the existence of a function that checks this relation is important for the construction of iterative morphological operators with an undefined number of iterations. For this kind of operator the satisfaction or not of the ordering relation between images produced in two consecutive iterations defines if the next iteration will be performed or not.

An image f_1 is *less than or equal* an image f_2 , denoted $f_1 \leq f_2$, if the following statement holds:

$$f_1 \leq f_2 \Leftrightarrow [f_1(x) \leq f_2(x), \text{ for any } x \in E], \quad (2)$$

where \leq denotes the less than or equal relation between integer numbers. Figures 8 and 9 show, respectively, a situation where this relation holds and where it does not.

An image f_1 is *equal* to an image f_2 , denoted $f_1 = f_2$, if the following statement holds:

$$f_1 = f_2 \Leftrightarrow [f_1(x) = f_2(x), \text{ for any } x \in E], \quad (3)$$

where $=$ denotes the equality relation between integer numbers.

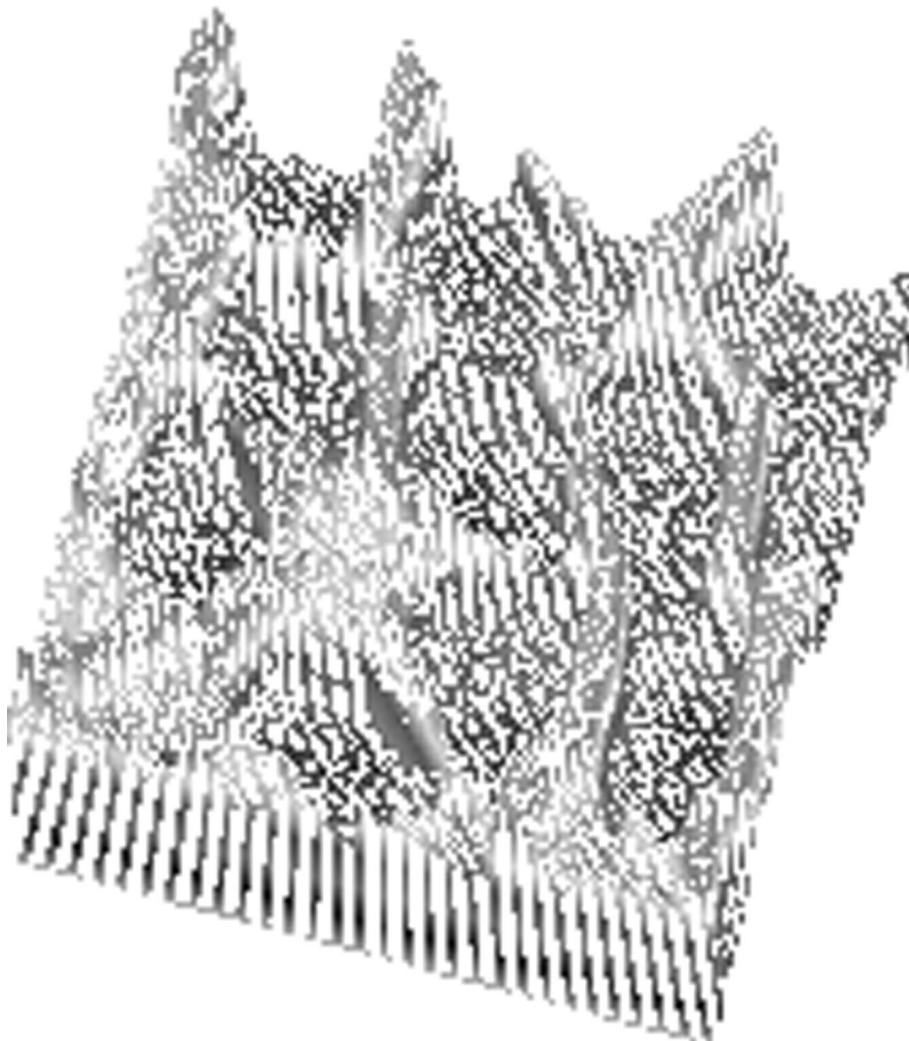


Fig. 7 3-D representation.

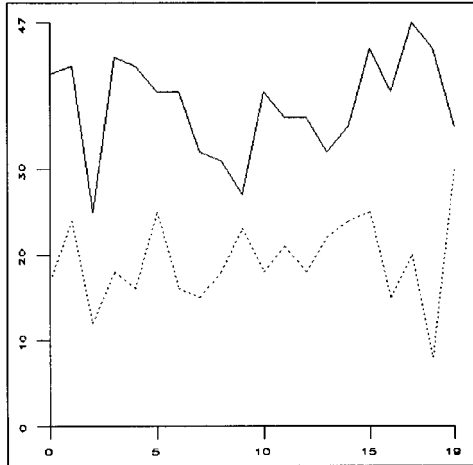


Fig. 8 $f_1 \leq f_2$.

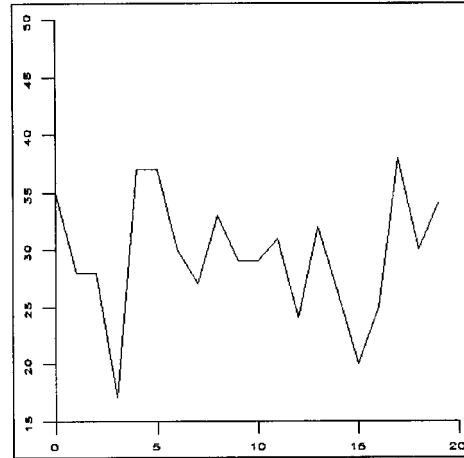


Fig. 10 Intersection.

2.3 Operations

Some useful definitions of punctual operations on images are given here. These definitions are based on structural properties of the interval $[0, k]$ of \mathbb{Z} .

The *intersection* of f_1 and f_2 , denoted $f_1 \wedge f_2$, is the function in K^E given by, for any $x \in E$,

$$(f_1 \wedge f_2)(x) = \min\{f_1(x), f_2(x)\}. \quad (4)$$

The *union* of f_1 and f_2 , denoted $f_1 \vee f_2$, is the function in K^E given by, for any $x \in E$,

$$(f_1 \vee f_2)(x) = \max\{f_1(x), f_2(x)\}. \quad (5)$$

Figures 10 and 11 show, respectively, the intersection and the union of the two 1-D functions of Figure 9.

The two binary operations \wedge and \vee from $K^E \times K^E$ to K^E are called *intersection* and *union*, respectively. Actually,

these operations applied to two distinct functions f_1 and f_2 produce, respectively, the infimum and the supremum of f_1 and f_2 , with respect to the partial ordering \leq . For this reason, the two binary operations \vee and \wedge are also called *infimum* and *supremum* operations (or simply infimum and supremum).

Usually, in practice, we need to perform successively several intersections or unions, so it is useful to have programs that implement extended intersections and extended unions.

Let $I = \{1, 2, 3, \dots, n\}$ be a set of indices. The *extended intersection* of a set of functions $\{f_i : i \in I\}$, denoted $\wedge\{f_i : i \in I\}$, is the function in K^E given by

$$\wedge\{f_i : i \in I\} = ((f_1 \wedge f_2) \wedge \dots) \wedge f_n. \quad (6)$$

The *extended union* of a set of functions $\{f_i : i \in I\}$, denoted $\vee\{f_i : i \in I\}$, is the function in K^E given by

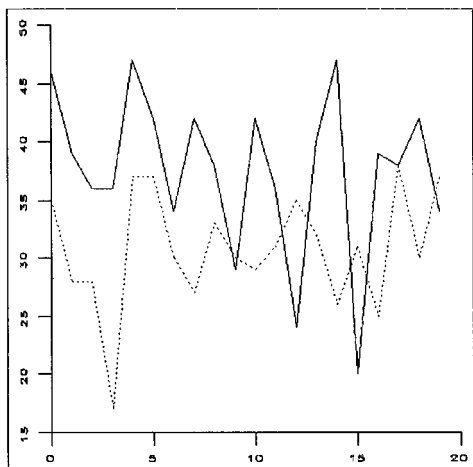


Fig. 9 $f_1 \not\leq f_2$.

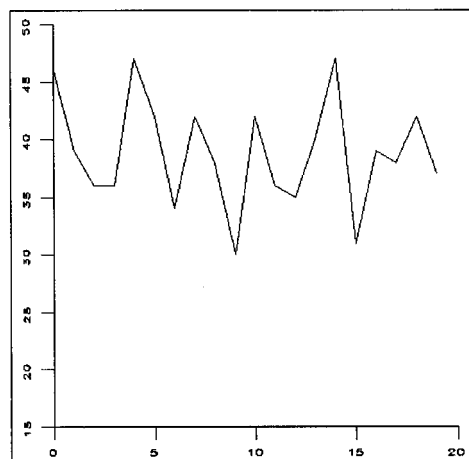


Fig. 11 Union.

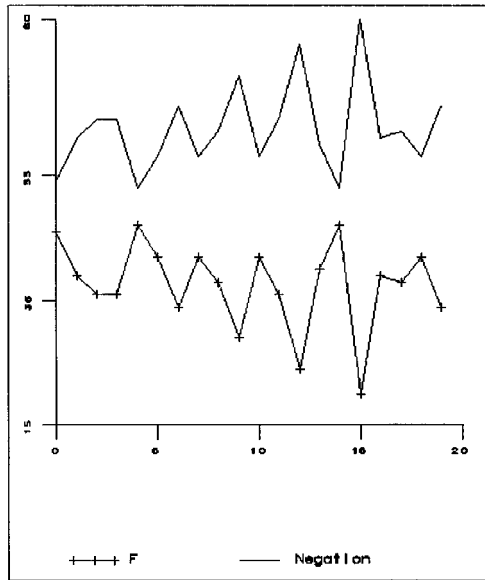


Fig. 12 Negation.

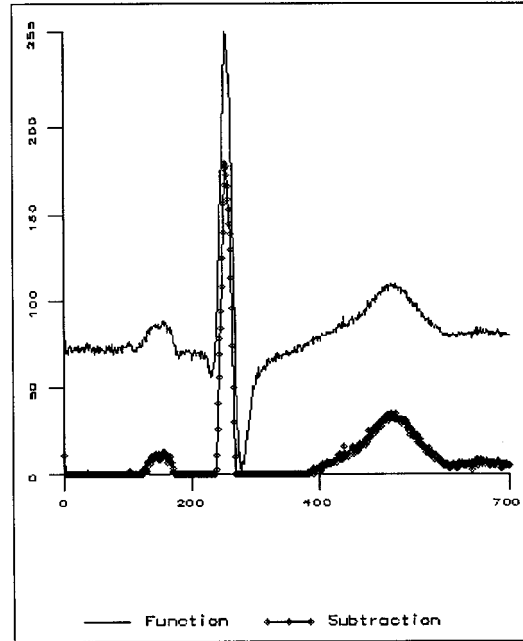


Fig. 13 Subtraction.

$$\vee\{f_i : i \in I\} = ((f_1 \vee f_2) \vee \dots) \vee f_n. \quad (7)$$

The *complementary* (or *negative*) of f , denoted $-f$, is the function in K^E given by, for any $x \in E$,

$$(-f)(x) = k - f(x). \quad (8)$$

The unary operation $-$ from K^E to K^E is called a *complementary operation* (or *negation*). Figure 12 shows a function and its negation.

The *difference* between f_1 and f_2 , denoted $f_1 - f_2$, is the function in K^E given by, for any $x \in E$,

$$(f_1 - f_2)(x) = \begin{cases} f_1(x) - f_2(x) & \text{if } f_2(x) \leq f_1(x) \\ 0 & \text{otherwise} \end{cases}. \quad (9)$$

The binary operation $-$ from $K^E \times K^E$ to K^E , is called *difference operation* (or *subtraction*). Actually, we have $f_1 - f_2 \leq f_1 \wedge (-f_2)$, and we get the equality for binary images, that is, for $f_1(E) = f_2(E) = \{0, k\}$. Figure 13 shows the subtraction of a constant from a function.

The *sum* of f_1 and f_2 , denoted $f_1 + f_2$, is the function in K^E given by, for any $x \in E$,

$$(f_1 + f_2)(x) = \begin{cases} f_1(x) + f_2(x) & \text{if } f_2(x) + f_1(x) \leq k \\ k & \text{otherwise} \end{cases}. \quad (10)$$

The binary operation $+$ from $K^E \times K^E$ to K^E is called *sum operation* (or *addition*). Figure 14 shows the addition of a constant to a function.

Let f , f_1 , and f_2 be such that $f_1 \leq f \leq f_2$. The *toggle* transform of f with respect to f_1 and f_2 , denoted $f_1[f]f_2$, is the function in K^E given by, for any $x \in E$,

$$(f_1[f]f_2)(x) = \begin{cases} f_1(x) & \text{if } (f - f_1)(x) \leq (f_2 - f)(x) \\ f_2(x) & \text{otherwise} \end{cases}. \quad (11)$$

The transform $[\cdot]$ from $K^E \times K^E \times K^E \times K^E$ to K^E is called the *toggle operator*.

Note that the toggle produces image enhancement, since it increases the relative distance between the gray levels of the image and preserves their partial ordering. Figure 15 shows the effect of the toggle operator, characterized by the two extreme functions of Figure 16, when applied to the intermediate function of Figure 16.

The *symmetrical difference* between f_1 and f_2 , denoted $f_1 \approx f_2$, is the function in K^E given by,

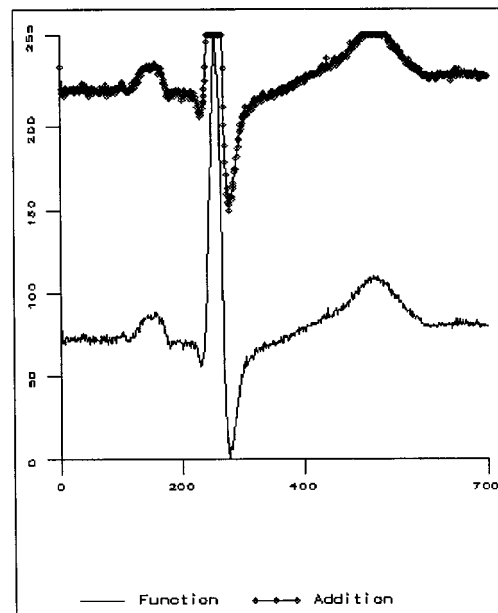


Fig. 14 Addition.

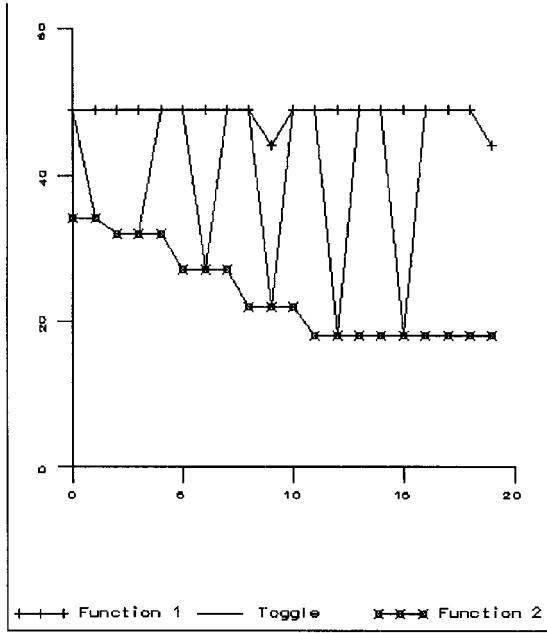


Fig. 15 Toggle.

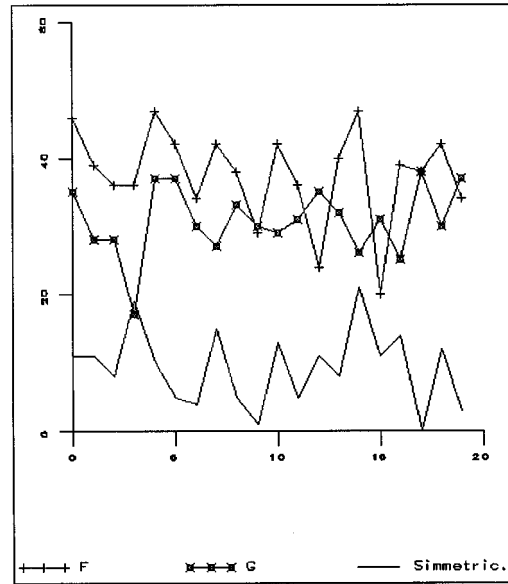


Fig. 17 Symmetrical difference.

$$(f_1 \approx f_2) = (f_1 - f_2) \vee (f_2 - f_1). \quad (12)$$

The transform \approx from $K^E \times K^E$ to K^E is called *symmetrical difference* operation.

Note that in the binary case the result of this operation is the set of points that is contained in one input image and not in the other. Figure 17 shows two functions and their symmetrical difference.

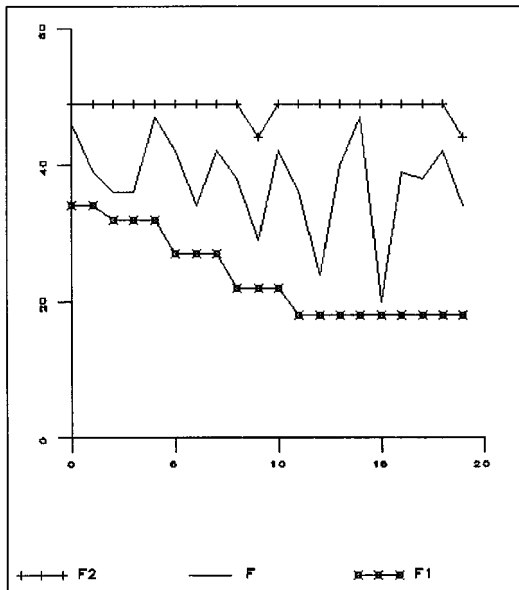


Fig. 16 Three ordered functions.

2.4 Operators and Operations on Operators

An *operator* is a mapping from K^E to K^E . We denote operators by Greek letters: $\alpha, \beta, \gamma, \Gamma$, etc. Therefore, an operator ψ transforms an image f into an image $\psi(f)$.

The *identity* operator is the mapping, denoted ι , given by, for any $f \in K^E$,

$$\iota(f) = f.$$

The operators may be combined by simple operations to build other operators. In this section, we present some operations defined on the space of operators.

Many useful operations on operators are inherited from operations on functions. In the following, we present some operations on operators inherited from the operations defined in the previous section. A common characteristic of all these operations is that they have a parallel nature.

Let ψ_1 and ψ_2 be two operators from K^E to K^E . The *intersection* of the operators ψ_1 and ψ_2 is the operator from K^E to K^E , denoted $\psi_1 \wedge \psi_2$, given by, for any $f \in K^E$,

$$(\psi_1 \wedge \psi_2)(f) = \psi_1(f) \wedge \psi_2(f).$$

The operation of *intersection* between the operators ψ_1 and ψ_2 , denoted \wedge , is the mapping given by

$$(\psi_1 \wedge \psi_2) \mapsto \psi_1 \wedge \psi_2.$$

Analogously, we define the operations of *union*, *addition*, and *subtraction*, denoted by \vee , $+$, and $-$, respectively. These binary operations can be extended to a sequence of identical operation or reduced to a unary operator by fixing one of its arguments. We write $\psi(f) \wedge g$ to denote the intersection of the functions $\psi(f)$ and g .

Let ψ be an operator from K^E to K^E . The *negation* of the operator ψ is the operator from K^E to K^E , denoted $-\psi$, given by, for any $f \in K^E$,

$$(-\psi)(f) = -\psi(f).$$

The operation of *negation* for operators, denoted $-$, is the mapping given by

$$\psi \mapsto -\psi.$$

An important operation on operators that is not inherited from an operation on functions is the composition. A salient difference between composition and the other operations presented before is that it has a sequential nature.

The *composition* of the operator ψ_1 with the operator ψ_2 is the operator from K^E to K^E , denoted $\psi_2\psi_1$, given by, for any $f \in K^E$,

$$(\psi_2\psi_1)(f) = \psi_2[\psi_1(f)].$$

The operation of *composition* of two operators is the mapping given by

$$(\psi_1, \psi_2) \mapsto \psi_2\psi_1.$$

Of course, any operator may be composed with itself. The succession of n self compositions of a generic operator ψ , where n is a positive integer, is denoted

$$\psi^n. \tag{13}$$

For example, the operator $\psi\psi$ will be denoted ψ^2 . This notation is extrapolated for $n=0$ by stating $\psi^0 = \iota$.

In practice, n self compositions of an operator ψ is implemented by a recursive procedure, that is, for any $f \in K^E$,

$$\psi^n(f) = f_n,$$

where $f_{i+1} = \psi(f_i)$ and $f_0 = f$.

Analogously, an undefined number of self compositions of an operator ψ is implemented by the following recursive procedure, for any $f \in K^E$,

$$\psi^\infty(f) = f_n,$$

where $f_{i+1} = \psi(f_i)$, $f_0 = f$, and n is the first iteration such that $f_n = f_{n+1}$. To implement this kind of iterative system it is necessary to use the relations stated in Section 2.2.

2.5 Structural Elements

Important classes of operators are characterized by functions from $B \subset \mathbb{Z}^2$ to K . These functions are called *structural elements* and are denoted by the lower case letter b .

2.5.1 Notations

Usually, the domain of a structural element is much smaller than the image that should be transformed. Hence, a convenient representation for it is a structure composed of a matrix and a vector (i.e., a pair of numbers), defined from the origin [i.e., the point (0,0)] to a point of the matrix, for example its right up corner. An example of a structural element is the pair

$$\begin{bmatrix} . & 1 & . \\ 5 & 4 & 5 \\ . & 8 & . \end{bmatrix}, (5,2),$$

where the dots in the matrix represent points that are out of the structural element domain. When the origin belongs to the domain of the structural element, it may be denoted just by a matrix, the origin being represented by a bold character. For example, the structural element

$$\begin{bmatrix} . & 1 & . \\ 5 & \mathbf{4} & 5 \\ . & 8 & . \end{bmatrix}, (1,1)$$

may be represented just by

$$\begin{bmatrix} . & 1 & . \\ 5 & \mathbf{4} & 5 \\ . & 8 & . \end{bmatrix}.$$

When a structural element is a function that is constantly equal to zero, it is called *flat*. We convention to substitute the function by its domain in the representation of a flat structural element and denote it by a capital letter. In this case, instead of representing the function by the gray level of its points, we represent it by a matrix of zeros (if the point is not in the function domain) and ones (if the point is in the function domain). For example, the flat structural element $\{(-1,-1), (0,0), (1,1)\}$ is represented by

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & \mathbf{1} & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

The image operators that are characterized by a structural element are denoted by a Greek letter with an index that may be the lower case letter b (for generic structural elements) or an upper case letter (for flat structural elements): $\gamma_b, \gamma_B, \phi_b, \phi_B, \Gamma_A, \Gamma_b, \dots$.

2.5.2 Discs, semisphere, and cylinder

In practical applications some important flat structural elements are the disks. The notion of disk depends on the notion of distance in a metric space.

A *distance* (or *metric*) d is any function defined from E to \mathbb{R} that, for any $u, v, w \in E$, satisfies the following properties

$$d(u, v) \geq 0 [d(u, v) = 0 \Leftrightarrow u = v], \tag{14}$$

$$d(u, v) = d(v, u), \tag{15}$$

$$d(u, w) \leq d(u, v) + d(v, w). \tag{16}$$

The value $d(u, v)$ is called the *distance* between u and v .

Let u and v be two elements of E represented by (u_1, u_2) and (v_1, v_2) , respectively. Three particular examples of distances are: the *Euclidean* distance

$$d_e(u,v)=[(u_1-v_1)^2+(u_2-v_2)^2]^{1/2}, \quad (17)$$

the *city block* distance

$$d_b(u,v)=|u_1-v_1|+|u_2-v_2|, \quad (18)$$

and the *chessboard* distance

$$d_m(u,v)=\max\{|u_1-v_1|, |u_2-v_2|\}. \quad (19)$$

Let x be an element of E and r be a positive integer. A *digital disk* under the distance d of center x and radius r is the subset of E given by

$$D_d(x,r)=\{y \in E: d(x,y) \leq r\}. \quad (20)$$

Two particular disks used often in practice are the 3×3 square centered at the origin and the cross contained in this square. These subsets are called the *elementary square* and the *elementary cross*.

Let $(o_1, o_2) \in \mathbb{Z}^2$ and $o_3, r \in K$. A *semisphere* of center (o_1, o_2, o_3) and radius r is the structural function b from B to K given by, for any $(x_1, x_2) \in B$,

$$b[(x_1, x_2)] = o_3 + [r^2 - (x_1 - o_1)^2 - (x_2 - o_2)^2]^{1/2}, \quad (21)$$

where B is the digital disk (under the Euclidean distance) of center (o_1, o_2) and radius r .

Let $h \in K$ and B be a digital disk (under the Euclidean distance). A *cylinder* is the structural function b from B to K given by, for any $x \in B$,

$$b(x) = h. \quad (22)$$

2.5.3 Operations on structural elements

Next, we present some operations on structural elements that are useful in characterizing important classes of operators.

Let W be a finite subset of \mathbb{Z}^2 and B be a subset of W . We denote by B^c the *complement* of B with respect to W , that is,

$$B^c = \{x \in W: x \notin B\}. \quad (23)$$

We denote by $B+u$ the *translate* of B by any vector u in \mathbb{Z}^2 , that is,

$$B+u = \{x+u: x \in B\}. \quad (24)$$

The Minkowski addition of two subsets A and B of \mathbb{Z}^2 is the subset $A \oplus B$ of \mathbb{Z}^2 , given by,

$$A \oplus B = \cup \{A+b: b \in B\}. \quad (25)$$

Of course, the two arguments of the Minkowski addition may be the same subset. The succession of $n-1$, where n is a positive integer, self additions of the subset B , denoted nB , is given by

$$nB = (B \oplus B) \oplus \dots \oplus B.$$

This notation is extrapolated for $n=0$ by stating

$$0B = \{(0, 0)\}.$$

We denote by \check{B} the *reflection* of B , that is,

$$\check{B} = \{-x: x \in B\}. \quad (26)$$

Another useful operation on structural elements is the clockwise integer rotation around the origin. Let (x_1, x_2) be an element of \mathbb{Z}^2 and let (x_{1a}, x_{2a}) denote the integer rotation of (x_1, x_2) by a degrees, that is, the point in \mathbb{Z}^2 that is the nearest neighbor of the real rotation of (x_1, x_2) by a degrees.

Let B be a flat structural element and a be a positive real number. The *integer rotation* of B by a degrees is the subset

$$B_a = \{(x_{1a}, x_{2a}): (x_1, x_2) \in B\}. \quad (27)$$

For example, the integer rotation by 45 degrees of the subset

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \text{ is the subset } \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

2.5.4 Interval

Some important families of operators are characterized by a collection of structural elements called intervals. In this subsection we introduce the notion of interval and the notations adopted to represent it.

Let W be a finite subset of \mathbb{Z}^2 . Let A and B be two subsets of W , such that $A \subset B$. The collection $[A, B]$ of subsets of W , such that

$$[A, B] = \{X \subset W: A \subset X \subset B\} \quad (28)$$

is called an interval of extremities A and B .

Often the interval $[A, B]$ is represented in a single matrix, whose values are 0 (when at a given point both A and B have the value 0), 1 (when at a given point both A and B have value 1) and \times (when at a given point A and B have different values). For example, the interval of extremities

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

is represented by

$$\begin{bmatrix} 1 & 1 & 1 \\ \times & 1 & \times \\ 0 & 0 & 0 \end{bmatrix}.$$

2.6 Dilations and Erosions

Dilations and erosions are the most fundamental classes of operators in mathematical morphology. Their algebraic definitions involve the notion of infimum and supremum on

complete lattices. Dilations commute with the supremum operation, while erosions commute with the infimum operation. In fact, through the Galois connection concept we can establish a one-to-one relation between the set of dilations and the set of erosions; that is, for each dilation there is a corresponding erosion and vice versa. We say that a pair formed by a dilation and its corresponding erosion constitutes an adjunction. These definitions and properties are quite general, but we study here just some subclasses of dilations and erosions defined on gray-scale images. For a complete characterization of translation invariant gray-level dilation and erosion, see Banon in Refs. 7 and 8.

A property of the elementary operators defined on gray-scale images is that any anti-dilation and anti-erosion can be obtained directly by the composition of, respectively, a dilation and an erosion with the negation. Thus, it is enough to state the notions of dilation and erosion to get the other corresponding elementary operators.

2.6.1 Dilations and erosions by structural elements

We now recall the definition of two important subclasses of dilations and erosions that are based on the Abelian group property of $(\mathbb{Z}^2, +)$.

Let $\dot{+}$ be the operation from $K \times \mathbb{Z}$ to K defined by, for any $t \in K$ and $v \in \mathbb{Z}$,

$$t \dot{+} v = \begin{cases} 0 & \text{if } t=0 \\ 0 & \text{if } t>0 \text{ and } t+v \leq 0 \\ t+v & \text{if } t>0 \text{ and } 0 \leq t+v \leq k \\ k & \text{if } t>0 \text{ and } t+v > k \end{cases} \quad (29)$$

Similarly, let $\dot{-}$ be the operation from $K \times \mathbb{Z}$ to K defined by, for any $t \in K$ and $v \in \mathbb{Z}$,

$$t \dot{-} v = \begin{cases} 0 & \text{if } t < k \text{ and } t-v \leq 0 \\ t-v & \text{if } t < k \text{ and } 0 \leq t-v \leq k \\ k & \text{if } t < k \text{ and } t-v > k \\ k & \text{if } t = k \end{cases} \quad (30)$$

The *dilation* of f by b is the function $\delta_b(f)$ in K^E , given by, for any $x \in E$,

$$\delta_b(f)(x) = \max\{f(y) \dot{+} b(x-y) : y \in (\check{B} + x) \cap E\}. \quad (31)$$

The *erosion* of f by b is the function $\varepsilon_b(f)$ in K^E , given by, for any x in E ,

$$\varepsilon_b(f)(x) = \min\{f(y) \dot{-} b(y-x) : y \in (B + x) \cap E\}. \quad (32)$$

The two operators δ_b and ε_b from K^E to K^E are called *dilation* and *erosion* by b .² In the previous expressions, we recall that $\max(\emptyset) = 0$ and $\min(\emptyset) = k$.

The fundamental idea under these definitions is to transform the image based on local comparisons between the image and translations of the structural element, which acts as a sensor of geometrical properties of the function. More precisely, $\varepsilon_b(f)(x)$ can be equivalently computed by translating spatially b until x , denoted b_x , and then translating b_x vertically by the maximum v , such that the translation of b_x by v is less or equal to f .

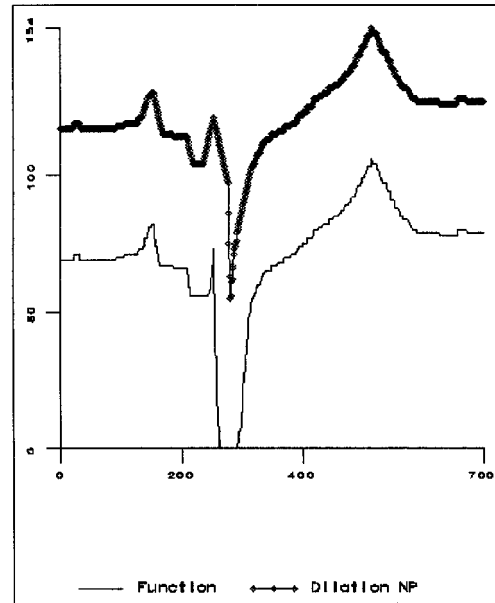


Fig. 18 1-D dilation.

The pair $(\varepsilon_b, \delta_b)$ forms an adjunction on K^E (Ref. 2, p. 388). Yet the use of the operations $\dot{+}$ and $\dot{-}$ is important to give similar treatment in the limit cases for the spatial and gray-scale dimensions: the spatial translation of the structural element is restricted to the image domain, while its vertical translation saturates at 0 and k . Figures 18 and 19 show a function and its dilation and erosion by the structural elements $g = [24 \ 25 \ \dots \ 46 \ 47 \ \mathbf{48} \ 47 \ 46 \ \dots \ 25 \ 24]$ and $g = [15 \ 16 \ \dots \ 28 \ 29 \ \mathbf{30} \ 29 \ 28 \ \dots \ 16 \ 15]$.

The dilation and erosion of a function f by a flat structural element b are, respectively, the functions $\delta_b(f)$ and $\varepsilon_b(f)$ in K^E , characterized just by the structural element

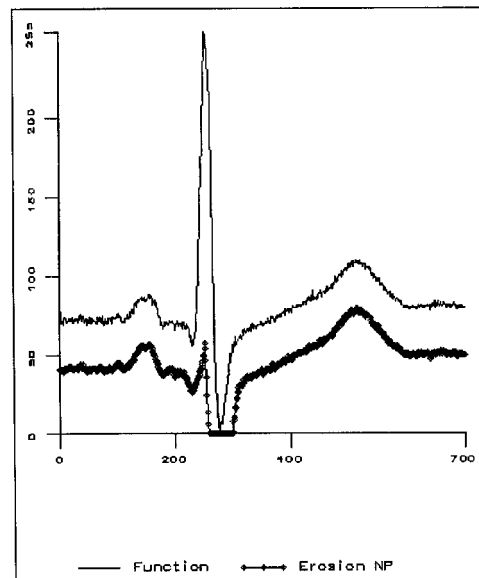


Fig. 19 1-D erosion.

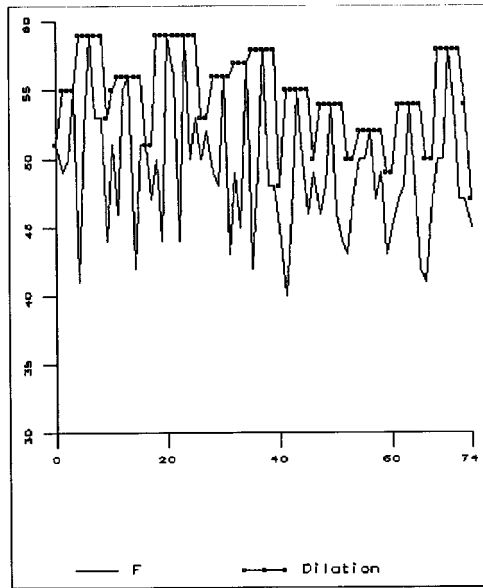


Fig. 20 1-D flat dilation.

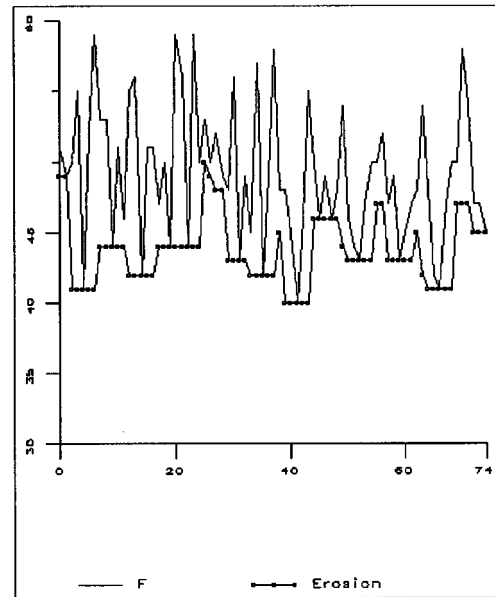


Fig. 21 1-D flat erosion.

B , the domain of b . The functions $\delta_B(f)$ and $\varepsilon_B(f)$ are called dilation and erosion of f by B and are given, for any $x \in E$, by

$$\delta_B(f)(x) = \max\{f(y) : y \in (\check{B} + x) \cap E\} \quad (33)$$

and

$$\varepsilon_B(f)(x) = \min\{f(y) : y \in (B + x) \cap E\}. \quad (34)$$

The two operators δ_B and ε_B from K^E to K^E are called dilation and erosion by B (Ref. 4, p. 80). Particularly, when $B = \{(0,0)\}$, the dilation and erosion by B are the identity operator.

Figures 20 and 21 show a function and its dilation and erosion by the flat structural element [11111]. Figures 22 and 23 show the dilation and erosion, respectively, of the binary image of Figure 24 by an Euclidean disc of diameter 10. Figure 25 shows a gray-scale image, while Figures 26 and 27 show its dilation and erosion by an Euclidean disc of diameter 10.

2.6.2 Self decomposition of dilations and erosions

Dilations and erosions can be decomposed in terms of other simpler dilations and erosions. Such decompositions have impact on the implementation of these operators.

If $(E, +)$ constitutes an Abelian group, then the dilations and erosions defined in the last section are translation invariant operators,⁵ i.e., to translate the image and then apply the operator is equivalent to applying the operator and then translating the resulting image.

Now we present two useful properties of dilations and erosions by flat structural elements.

Property 1. The dilation (erosion) by a structural set B is equivalent to the union (intersection) of dilations (erosions) by subsets in a family whose union is B , that is,



Fig. 22 Binary dilation.



Fig. 23 Binary erosion.

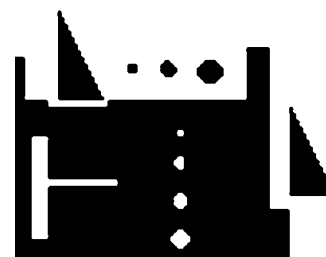


Fig. 24 Binary image.

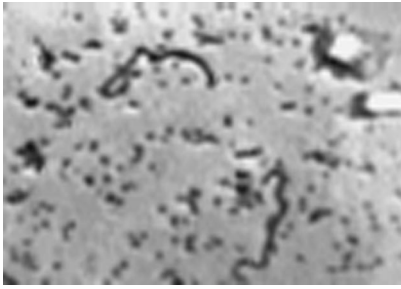


Fig. 25 Gray-scale image. Transmitters of filariasis.

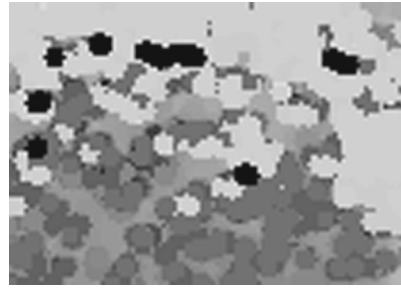


Fig. 27 Gray-scale erosion.

$$\delta_B = \vee \{ \delta_{B_i} : \cup B_i = B \}, \tag{35}$$

$$(\varepsilon_B = \wedge \{ \varepsilon_{B_i} : \cup B_i = B \}). \tag{36}$$

Property 2. If the dilation (erosion) by the Minkowski addition of two subsets A and B is a translation invariant operator, then it is equivalent to the composition of the dilation (erosion) by A and B , that is,

$$\delta_{A \oplus B} = \delta_A \delta_B, \tag{37}$$

$$(\varepsilon_{A \oplus B} = \varepsilon_A \varepsilon_B). \tag{38}$$

A particular consequence of Properties 1 and 2 is that dilation and erosion by any subset B can be built by composing dilations and erosions by subsets of the elementary square. Some studies reveal that this decomposition can lead to algorithms for dilations and erosions more efficient than the direct ones (Ref. 9, p. 48).

The two operators δ_b^n and ε_b^n , where n is a nonnegative integer, from K^E to K^E are called, respectively, n -dilation and n -erosion. Figures 18 and 19 show a function and its 24 dilation and 15 erosion by the structural element $[1 \ 2 \ 1]$. Actually, when b is flat, δ_b^n and ε_b^n are equivalent to the dilation and erosion by nB .

We call the subsets

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

of directional structural elements.

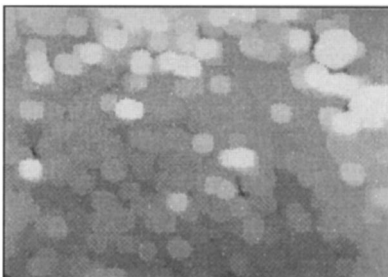


Fig. 26 Gray-scale dilation.

Let L be a generic digital line segment (i.e., that has an arbitrary size and slope) with one of its extremities at the origin. The segment L can be represented as a sequence of directional structural elements B_1, B_2, \dots, B_n , which describes the path to go from the origin to the other extremity of L . Hence, as a consequence of Proposition 1, the dilation and erosion of f by the line segment L are given by

$$\begin{aligned} \delta_L(f) = & \iota \vee \delta_{B_1}(f) \vee \delta_{B_2}(\delta_{B_1}(f)) \\ & \vee \dots \vee \delta_{B_n}(\delta_{B_{n-1}}(\dots(\delta_{B_1}(f)))) \end{aligned} \tag{39}$$

and

$$\begin{aligned} \varepsilon_L(f) = & \iota \wedge \varepsilon_{B_1}(f) \wedge \varepsilon_{B_2}(\varepsilon_{B_1}(f)) \\ & \wedge \dots \wedge \varepsilon_{B_n}(\varepsilon_{B_{n-1}}(\dots(\varepsilon_{B_1}(f)))) \end{aligned} \tag{40}$$

2.6.3 Distance function and threshold

Two other important families of erosions are the distance functions and the thresholds. A distance function maps a binary image into an equivalent gray-scale model, while a threshold transforms a gray-scale image into a simplified binary model.

Let x be an element of E , X be a subset of E , and d be a distance. The distance between the point x and the subset X , under the distance d , is the value given by

$$d(x, X) = \min \{ d(x, y) : y \in X \}. \tag{41}$$

Let k be a positive integer such that

$$k > \max \{ d(x, y) : x, y \in E \}. \tag{42}$$

The distance function applied to f , under the distance d , is the function in K^E given by, for any $x \in E$,

$$\Psi_d(f)(x) = d(x, \{ y \in E : f(y) = 0 \}). \tag{43}$$

The operator Ψ_d from $\{0, k\}^E$ to K^E is called the distance function operator under the distance d . The distance functions are erosions from $\{0, k\}^E$ to K^E .

Figures 28 and 29 show a binary image and its corresponding distance function. Note that in the image representation of the distance function used in this figure, higher points in the function surface are represented by darker gray levels.

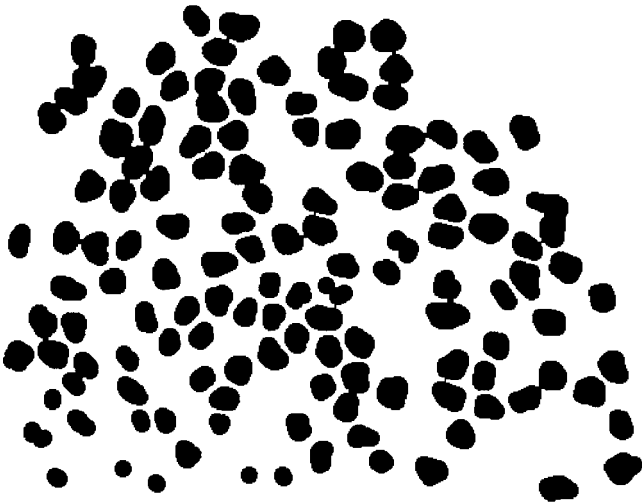


Fig. 28 Overlapped blobs.

The comparison between f_1 and f_2 , denoted $f_1 \leq f_2$, is the function in K^E given by, for any $x \in E$,

$$(f_1 \leq f_2)(x) = \begin{cases} k & \text{if } f_1(x) \leq f_2(x) \\ 0 & \text{otherwise} \end{cases} \quad (44)$$

The unary operations $\cdot \leq f$ and $f \leq \cdot$ from K^E to K^E are called, respectively, *right* and *left adaptive threshold* with respect to f . These unary operations are an antidilation and an erosion, respectively.

The composition of a distance function Ψ_d with the left adaptive threshold operator $f \leq \cdot$ is an erosion in $\{0, k\}^E$. If f is a constant function equal to $r+1 \in K$, then $f \leq \Psi_d$ is the erosion characterized by the digital disk of center $(0,0)$ and radius r , that is,

$$\varepsilon_{D_d}((0,0), r) = (r+1 \leq \Psi_d). \quad (45)$$

2.7 Geometric Manipulation

In this section, we present some elementary operators that change the domain of definition of the image by expansion



Fig. 29 Distance function.

or shrinking. The difference between these operators and the conventional expansion and shrink is that in the conventional procedures just one point is considered, while in the morphological procedures a neighborhood around each point is taken into account. The analysis of the neighborhood follows the one performed by dilation and erosion.

Let \mathbb{N} be the set of positive integers and let \mathbb{N}^* be the set of nonnegative integers. Let $(n_1, n_2) \in \mathbb{N}^2$ and let $E = [0, 1, \dots, n_1 - 1] \times [0, 1, \dots, n_2 - 1]$ be a rectangle of \mathbb{Z}^2 . Let $s = (s_1, s_2) \in \mathbb{N}^2$ and let $o = (o_1, o_2) \in \mathbb{N}^{*2}$, such that $o_1 < s_1$ and $o_2 < s_2$. The vectors s and o will be called scale factor and offset.

Let $sE = [0, 1, \dots, (s_1 n_1) - 1] \times [0, 1, \dots, (s_2 n_2) - 1]$ and let $B \subset sE \oplus (sE)$. sE will be the domain of the expanded images.

Let $f \in K^E$, we define f and \bar{f} in K^{sE} by, for any $y \in sE$,

$$\underline{f}(y) = \begin{cases} f[(y-o)/s] & \text{if } \exists x \in E: y = sx + o \\ 0 & \text{otherwise} \end{cases}, \quad (46)$$

$$\bar{f}(y) = \begin{cases} f((y-o)/s) & \text{if } \exists x \in E: y = sx + o \\ k & \text{otherwise} \end{cases}. \quad (47)$$

The functions \underline{f} and \bar{f} are expansions of f with trivial interpolations. In the first function, the interpolated value is 0, while in the second function it is k . The operators $\dot{+}$ and $\dot{-}$ from K^E to K^{sE} are called, respectively, inf and sup expansion.

The expansion by dilation of f in K^E by the structural element B , the scale factor s , and offset o , is the function $\text{expand} - \delta_{B,s,o}(f)$ in K^{sE} , given by, for any $x \in sE$,

$$\text{expand} - \delta_{B,s,o}(f)(x) = \max\{f(y) \dot{+} b(x-y) : y \in (\bar{B} + x) \cap sE\}. \quad (48)$$

The expansion by erosion of f in K^E by the structural element B , the scale factor s , and offset o , is the function $\text{expand} - \varepsilon_{B,s,o}(f)$ in K^{sE} , given by, for any $x \in sE$,

$$\text{expand} - \varepsilon_{B,s,o}(f)(x) = \min\{\bar{f}(y) \dot{-} b(y-x) : y \in (B + x) \cap sE\}. \quad (49)$$

The two operators $\text{expand} - \delta_{B,s,o}$ and $\text{expand} - \varepsilon_{B,s,o}$ from K^E to K^{sE} are called *expansion by dilation* and *expansion by erosion*, with the structural element B , the scale factor s , and offset o .

Note that the expansions by dilation and erosion are a composition of an expansion with a dilation and an erosion. As 0 and k are the neutral elements for dilation and erosion the sup expansion is used in the first case, while the inf expansion is used in the second one.

Figures 30 and 31 show an image and its expansion by dilation, using as structural element the image of Figure 30 itself.



Fig. 30 Image to be expanded.

Let $E/s = [0, 1, \dots, \lfloor n_1/s_1 \rfloor - 1] \times [0, 1, \dots, \lfloor n_2/s_2 \rfloor - 1]$, where $\lfloor x \rfloor$ is the integer part of x , and let $B \subset E \oplus \check{E}$, where E/s is a shrinking of E .

The *shrinking by dilation* of f in K^E by the structural element B , the scale factor s , and offset o is the function $\text{shrin} - \delta_{B,s,o}(f)$ in K^{sE} , given by, for any $x \in E/s$,

$$\text{shrin} - \delta_{B,s,o}(f) = \max\{f(y) \div b(x - y) : y \in (\check{B} + sx + o) \cap E\}. \quad (50)$$

The *shrinking by erosion* of f in K^E by the structural element B , the scale factor s , and offset o is the function $\text{shrin} - \varepsilon_{B,s,o}(f)$ in K^{sE} , given by, for any $x \in E/s$,

$$\text{shrin} - \varepsilon_{B,s,o}(f)(x) = \min\{f(y) \div b(y - x) : y \in (B + sx + o) \cap E\}. \quad (51)$$

The two operators $\text{shrin} - \delta_{B,s,o}$ and $\text{shrin} - \varepsilon_{B,s,o}$ from K^E to K^E are called, respectively, *shrinking by dilation* and *shrinking by erosion*, with the structural element B , the scale factor s , and offset o .

The shrinking by dilation and erosion correspond to an evaluation of the dilation and erosion operators on a grid of E followed by a sampling.

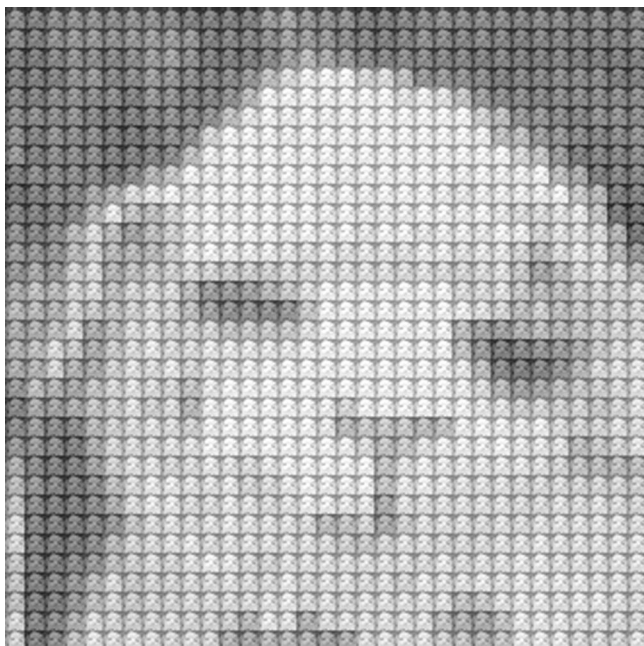


Fig. 31 Expansion by dilation.

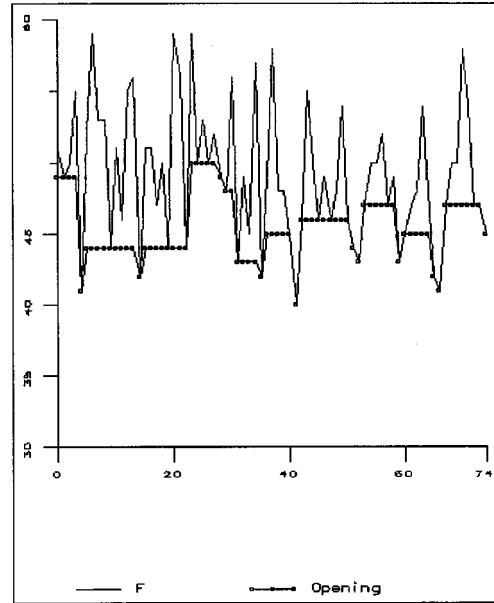


Fig. 32 Flat opening.

2.8 Morphological Filters

As stated by Serra (Ref. 1, p. 101), the morphological filters are the family of increasing (i.e., preserve the partial ordering) and idempotent (i.e., a second application of the operator does not affect the result) operators. These filters are useful for shape and size classification as well as for noise reduction.

In this section, we present some morphological filters that have been used for image processing. All of these operators are presented in dual pairs. The behavior of the dual of a filter is equivalent to the behavior of the filter itself when applied to the complementary image; that is, if a filter affects the peaks of the images, then its dual affects similarly the valleys and vice-versa.

Two important morphological filters are opening and closing. They are particularly useful for shape and size classification and constitute the basis for a complete family of filters with a rich set of properties.

The operators γ_b and ϕ_b from K^E to K^E , given by

$$\gamma_b = \delta_b \varepsilon_b \quad (52)$$

and

$$\phi_b = \varepsilon_b \delta_b, \quad (53)$$

are called (morphological) *opening* and *closing* by b (Ref. 3, p. 50).

Figures 32 and 33 show a function and its opening and closing by the flat structural element $[11111]$. Figures 34 and 35 show a function and its opening and closing by the structural element $[5 \ 10 \ 5]$.

The operators $\gamma_{n,b}$ and $\phi_{n,b}$ from K^E to K^E , given by

$$\gamma_{n,b} = \delta_b^n \varepsilon_b^n \quad (54)$$

and

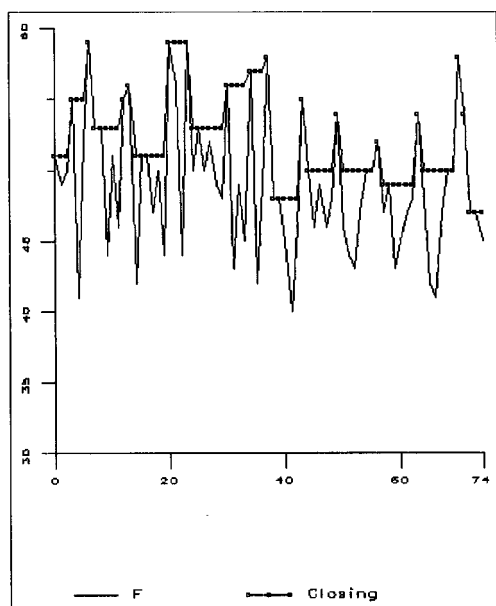


Fig. 33 Flat closing.

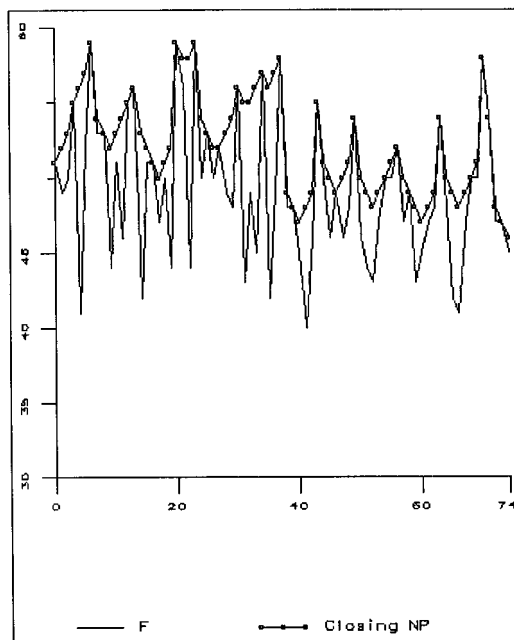


Fig. 35 Closing.

$$\phi_{n,b} = \varepsilon_b^n \delta_b^n, \quad (55)$$

are called n -opening and n -closing by b . Actually, when b is flat, $\gamma_{n,B}$ and $\phi_{n,B}$ are equivalent to the opening and closing by nB .

The shape and size classification based on directional properties may be performed by the following specialized morphological operators.

Let L be a digital line segment. The operator γ_L and ϕ_L from K^E to K^E , given by

$$\gamma_L = \delta_L \varepsilon_L \quad (56)$$

and

$$\phi_L = \varepsilon_L \delta_L, \quad (57)$$

are called *opening* and *closing by the line segment L* .

The next operators are built by sequential compositions of openings and closings. In fact, this family of operators cover all possible sequential combination of openings and closings. Contrasting with the opening and closing operators, they have similar actions on the peaks and valleys of the images. However, there are still some differences of behavior due to the fact of beginning by an opening or by a closing. The operators that begin by an opening affect more the peaks, while the ones that begin by a closing affect more the valleys.

The two operators $\theta_{n,b}$ and $\psi_{n,b}$ from K^E to K^E , given by

$$\theta_{n,b} = \phi_{n,b} \gamma_{n,b} \quad (58)$$

and

$$\psi_{n,b} = \gamma_{n,b} \phi_{n,b}, \quad (59)$$

are called n - $\phi\gamma$ -filter and n - $\gamma\phi$ -filter by b (Ref. 1, p. 203).

The two operators $\theta_{n,b}$ and $\psi_{n,b}$ from K^E to K^E , given by

$$\theta_{n,b} = \gamma_{n,b} \phi_{n,b} \gamma_{n,b} \quad (60)$$

and

$$\psi_{n,b} = \phi_{n,b} \gamma_{n,b} \phi_{n,b}, \quad (61)$$

are called n - $\phi\gamma\phi$ -filter and n - $\gamma\phi\gamma$ -filter by b (Ref. 1, p. 203).

Alternated sequential filters consist of iterations of filters of increasing strongness. They produce refined filter effects by modifying sequentially details of increasing scale.

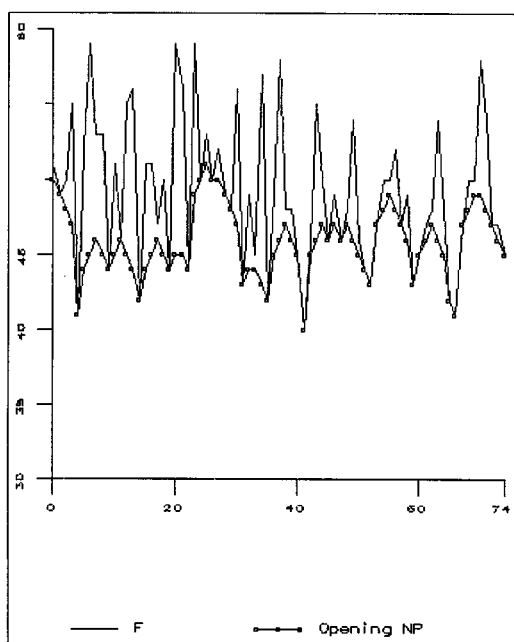


Fig. 34 Opening.

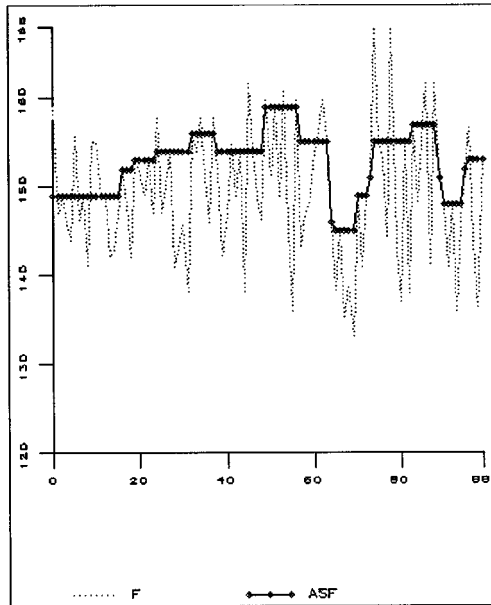


Fig. 36 Flat ASF filter.

Denote the $n - \phi\gamma$, $n - \gamma\phi$, $n - \phi\gamma\phi$, $n - \gamma\phi\gamma$ filter by b , generically by $\psi_{n,b}$. The operator $\psi_{n,b}$ from K^E to K^E , given by

$$\psi_{n,b} = \psi_{n,b} \psi_{n-1,b} \dots \psi_{1,b}, \quad (62)$$

is called an n alternated sequential filter by b (Ref. 1, p. 203).

Figure 36 shows a function and its filtering by an alternated sequential filter of type $3 - \phi\gamma$, characterized by the flat structural element [111]. Figure 37 shows the result of the application of an alternated sequential filter on the image of Figure 38.

Another interesting morphological filter is the center filter.



Fig. 37 Filtered image.

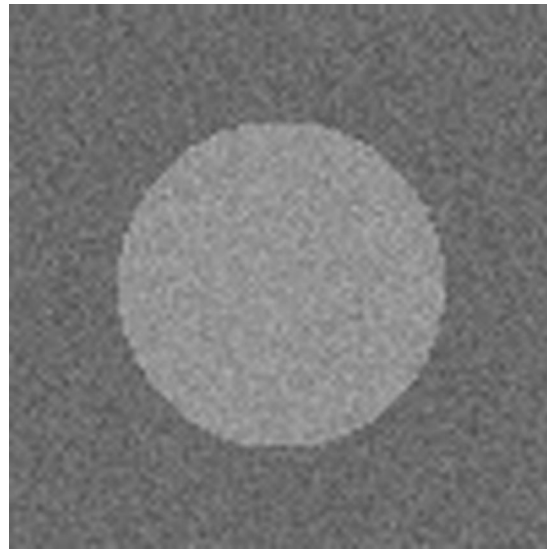


Fig. 38 Corrupted image.

Its remarkable property is that it is self-dual, that is, it affects peaks and valleys of the images exactly in the same way.

The operator β_b , from K^E to K^E , given by

$$\beta_b = (\iota \wedge \phi_b \gamma_b \phi_b) \vee \gamma_b \phi_b \gamma_b, \quad (63)$$

is called the *primitive of the center filter*. The infinite successive compositions of this primitive β_b^∞ is called the *center filter* (Ref. 1, p. 159).

Yet several other morphological filters could be built from the filters that we have just presented. For example, a quite useful refinement of opening and closing operators is built by the union of openings and the intersection of closings. Heijmans presented in Ref. 10 other strategies to build new morphological filters from the ones that we have presented here.

2.9 Connected Filters

Connected filters constitute a particular family of morphological filters that are useful to change topological properties of images (to know more about digital topology, see Ref. 11). On binary images, the effect of connected filters is to eliminate objects of the image or of its background (i.e., holes). On gray-scale images, the effect of the connected filters is to join adjacent flat zones (i.e., plateaus of constant gray level). On binary images these filters are based on the connectivity of the 2-D space and on gray-scale images they are based on the connectivity of the 2-D or 3-D spaces.

This family of filters is particularly useful in image segmentation, because connected filters have good properties for selecting objects preserving the details of their edges.

Let g be an element of K^E . The operators $\delta_{b,g}$ and $\varepsilon_{b,g}$ from K^E to K^E , given by

$$\delta_{b,g} = \delta_b \wedge g \quad (64)$$

and



Fig. 39 Erosion by a disk.

$$\varepsilon_{b,g} = \varepsilon_b \setminus g, \quad (65)$$

are called *conditional* (or *geodesic*) *dilation* and *erosion* by b given g (Ref. 3, p. 393).

The operators $\delta_{b,g}^n$ and $\varepsilon_{b,g}^n$, where n is a nonnegative integer, are called *n-conditional dilatation* and *erosion* by b given g .

Let f be an element of K^E . The operators $\gamma_{b,f}$ and $\phi_{b,f}$ from K^E to K^E , given by, for any $g \in K^E$,

$$\gamma_{b,f}(g) = \delta_{b,g}^\infty(f) \quad (66)$$

and

$$\phi_{b,f}(g) = \varepsilon_{b,g}^\infty(f), \quad (67)$$

are called *inf-reconstruction* and *sup-reconstruction* from the marker f .¹²

Figure 39 shows an erosion of the image of Figure 28. Figure 40 shows the inf-reconstruction of the image of Figure 28 from the image of Figure 39, using the elementary

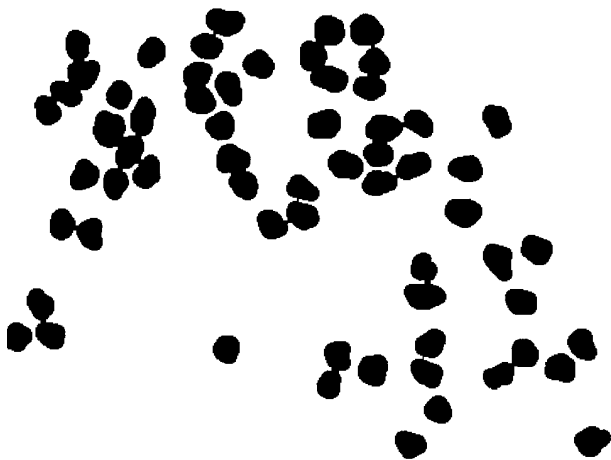


Fig. 40 Inf-reconstruction.

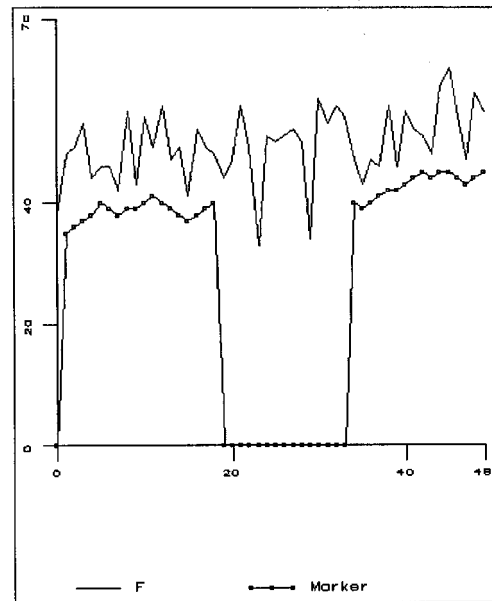


Fig. 41 Function and its marker.

square as the flat structural element. Note that the small objects were eliminated by this filtering and the remaining objects keep their original shapes.

The inf-reconstruction using the erosion of the input image as a marker is also called opening by reconstruction. The reason for this name comes from two facts: it is built by an inf-reconstruction procedure, and it is an algebraic opening, that is, it belongs to the class of operators that are increasing, anti-extensive (the transformed image is always less or equal the input image), and idempotent. Other examples of operators that are also algebraic openings are the morphological openings and the union of morphological openings.

Dually, the sup-reconstruction that use a dilation as a marker is called closing by reconstruction, since it is an element of the class of the algebraic closing; that is, the class of operators that are increasing, extensive (the input image always is less or equal the transformed image), and idempotent. Other examples of operators that are algebraic closings are the morphological closings and the intersection of morphological closings.

Figure 41 shows a function and a marker. Figure 42 shows the same function and its inf-reconstruction from the marker, using the subset [111] as the flat structural element. Figure 43 shows the sup-reconstruction of the image of Figure 25 from a dilation by a large structural element (closing by reconstruction), using the elementary square as the flat structural element.

Figure 44 shows a function and a marker. Figure 45 shows the inf-reconstruction of the function from the marker, using as structural element (nonflat) the function [111]. Note that the mountains pointed by the marker were preserved, while the others were eliminated.

The application of the reconstruction operators with good markers gives a powerful tool for image segmentation. Some operators derived from these operators are: clos-

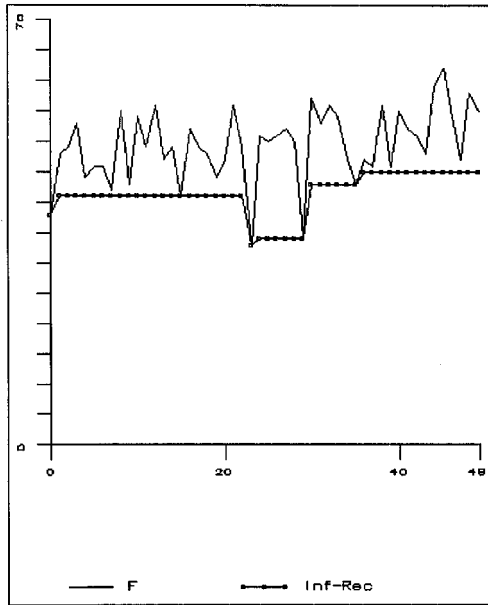


Fig. 42 Flat inf-reconstruction.

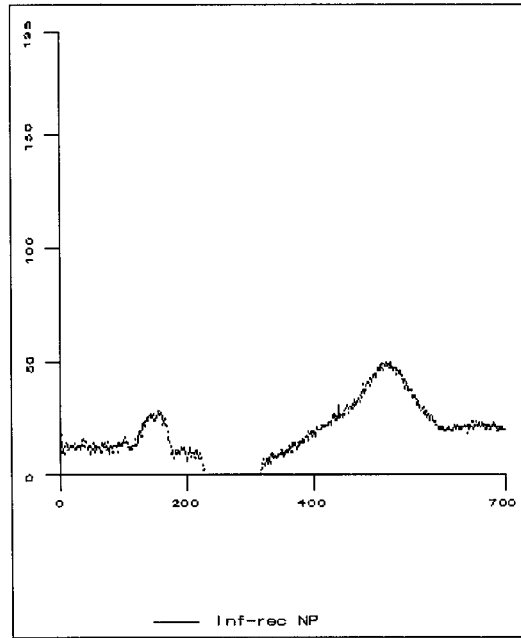


Fig. 45 Inf-reconstruction.

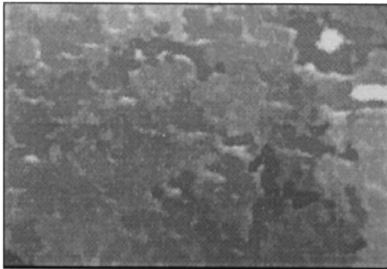


Fig. 43 Closing by reconstruction.

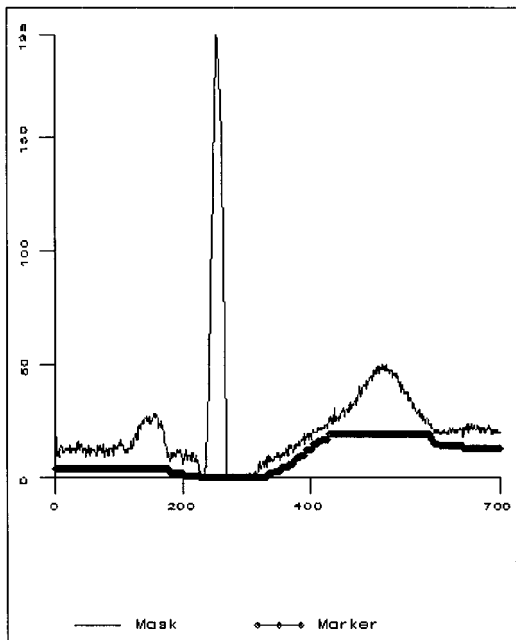


Fig. 44 Function and its marker.

ing of holes, labeling, minima imposition, and regional minimum and maximum.

All these operators depend on the choice of a connectivity relation and this is done via the choice of a flat structural element B . Usually, the element B adopted is the elementary square (8 connectivity) or the elementary cross (4 connectivity).

The operator Φ_B from $\{0,k\}^E$ to $\{0,k\}^E$, given by, for any $g \in \{0,k\}^E$,

$$\Phi_B(g) = -\gamma_{B,f}(-g), \tag{68}$$

is called the *closing of holes* in g . Note that in $\Phi_B(g)$ there are no holes. The image of Figure 46 is the effect of the application of this operator on the image of Figure 47.

Let $k \geq |E|$, let $i \mapsto x_i$ be a numbering process of the elements of E (that is a bijection from $[1, \dots, |E|] \subset \mathbb{N}$ to E), and let f be an element of K^E such that $f(x_i) = i$, for $x_i \in E$. The operator Λ_B from $\{0,k\}^E$ to K^E , given by, for any $g \in \{0,k\}^E$,

$$\Lambda_B(g) = \gamma_{B,g \wedge f}(g), \tag{69}$$

is called *labeling* of g (Ref. 3, p. 405). Note that in $\Lambda_B(g)$ each point of a connected component of g is associated to the same value.

The labeling operator is fundamental for applications that depend on geometrical measures of the objects.

Let f be an element of $\{0,k\}^E$. The operator $\Gamma_{f,B}$ from K^E to K^E , given by, for any $g \in K^E$,

$$\Gamma_{f,B}(g) = \phi_{B,f}((g \vee 1) \wedge (k-1) \wedge f), \tag{70}$$

is called *minima imposition* on g based on the marker f .¹²

This filter is quite important in image segmentation. For example, if it is applied on an image of enhanced edges and

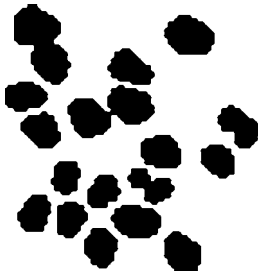


Fig. 46 Closing of holes.

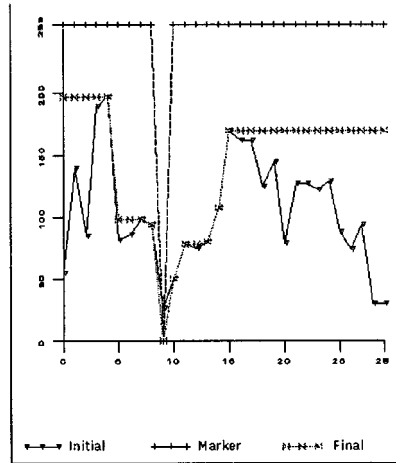


Fig. 48 Minima imposition.

the marker image has a connected component pointing each object of interest for the segmentation, it will eliminate all the undesired edges and keep just the true edges of the chosen objects. Figure 48 shows a function, a marker, and the minima imposition on the function based on the marker.

A point x in E is a *local maximum* (resp. *local minimum*) of a function $f \in K^E$ if and only if the value $f(x) \geq f(y)$ [resp., $f(x) \leq f(y)$], for any y in the neighborhood of x . Equivalently, we can say that the local maximum (resp., local minimum) are the invariant points of the dilation (resp., erosion), that is, the set of points where the function $f \equiv \delta_B(f)$ [resp., $f \equiv \varepsilon_B(f)$] value is k . Note that \equiv is the equality operator defined by formula 84.

A *regional maximum* (resp., *regional minimum*) M of a function $f \in K^E$ is a connected component with a given value h (plateau of altitude h), such that every point in the neighborhood of M has a strictly lower (resp., higher) value. The regional maximum and the regional minimum can be extracted from the functions by the following operators.

The operators ϱ_B^{\max} and ϱ_B^{\min} from K^E to $\{0, k\}^E$, given by, respectively,

$$\varrho_B^{\max}(f) = (1 \equiv ((f+1) - \gamma_{B,f}(f+1))) \vee (f \equiv k) \quad (71)$$

and

$$\varrho_B^{\min}(f) = (1 \equiv (\phi_{B,f}(f-1) - (f-1))) \vee (0 \equiv f), \quad (72)$$

are called *regional maximum operator* and *regional minimum operator*.¹³

The holes in the image of Figure 49 are the regional maxima of the image of Figure 29.

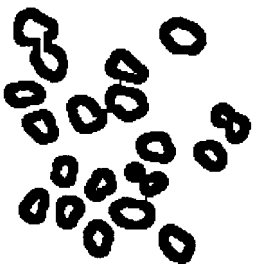


Fig. 47 Blob contours.

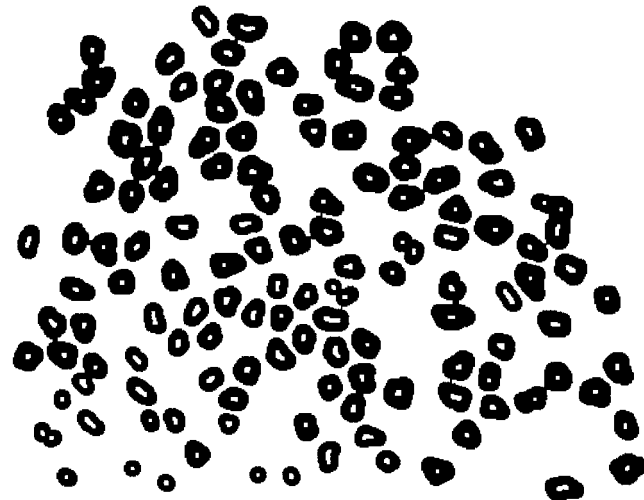


Fig. 49 Objects and their markers.

2.10 Residues

The family of residue operators consist of operators that are built by the difference between two morphological operators. They are commonly used to detect changes of contrast, to segment and to compress images.

2.10.1 Gradient

Let the flat structural elements A and B be subsets of the elementary square. The operator $\Psi_{A,B}$ from K^E to K^E , given by

$$\Psi_{A,B} = \delta_A - \varepsilon_B, \quad (73)$$

is called (morphological) *gradient* (Ref. 3, p. 437). This operator performs the enhancement of edges. Particularly, if A and B are two or three point line segments it does a directional enhancement of edges. The image of Figure 50 is the complement of the gradient (with A and B being the elementary square) of the image of Figure 51.

2.10.2 Top hats and reconstruction residues

An important class of residue operators are the top hats. These operators are built by the subtraction of an opening



Fig. 50 Inverted gradient.

from the identity or of the identity from a closing. Note that these definitions are consistent with the fact that openings are antiextensive operators and closings are extensive operators. In the following, we give some top hat operators defined from the morphological and reconstruction opening and closing.

The operator γ_b^t from K^E to K^E , given by

$$\gamma_b^t = \iota - \gamma_b, \tag{74}$$

is called (morphological) *opening top hat* by b (Ref. 3, p. 474).

The operator ϕ_b^t from K^E to K^E , given by

$$\phi_b^t = \phi_b - \iota, \tag{75}$$

is called (morphological) *closing top hat* by b (Ref. 3, p. 474).

Let A be a subset of the elementary square. The operator $\Gamma_{A,b}^t$ from K^E to K^E , given by, for any $g \in K^E$,



Fig. 51 Calculator pad.

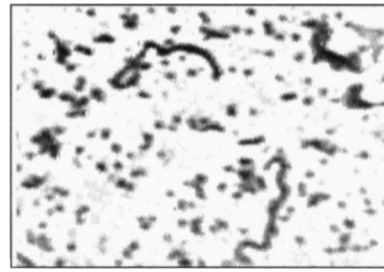


Fig. 52 Subtraction.

$$\Gamma_{A,b}^t(g) = g - \gamma_{A,\epsilon_b(g)}(g), \tag{76}$$

is called *opening by reconstruction top hat* by A and b (Ref. 3, p. 474).

The operator $\Phi_{A,b}^t$ from K^E to K^E , given by, for any $g \in K^E$,

$$\Phi_{A,b}^t(g) = \phi_{A,\delta_b(g)}(g) - g, \tag{77}$$

is called *closing by reconstruction top hat* by A and b (Ref. 3, p. 474).

The image of Figure 52 is the result of the application of the closing by reconstruction top hat on the image of Figure 25.

The next residue operator uses the inf-reconstruction from a particular marker, the frame image, to eliminate objects that are on the frame.

The operator Γ_B from $\{0,k\}^E$ to $\{0,k\}^E$, given by,

$$\Gamma_B = \iota - \gamma_{B,f}, \tag{78}$$

where f is the frame of E , is called *objects of g on frame off*. Note that in $\Gamma_B(g)$ there are no connected components on the frame of E . The image of Figure 46 is the result of the application of this operator to the image of Figure 53.

2.10.3 Morphological skeletons

In the following, we present three operators that are based on the detection of centers of maximal shapes contained in an object (Ref. 9, p. 166). These operators are built by the union of particular residue operators.

The operator σ_B from $\{0,k\}^E$ to $\{0,k\}^E$, given by

$$\sigma_B = \bigvee \{ \epsilon_B^i - \gamma_B \epsilon_B^i : i = 0, 1, \dots \}, \tag{79}$$

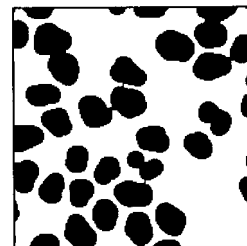


Fig. 53 Blobs.

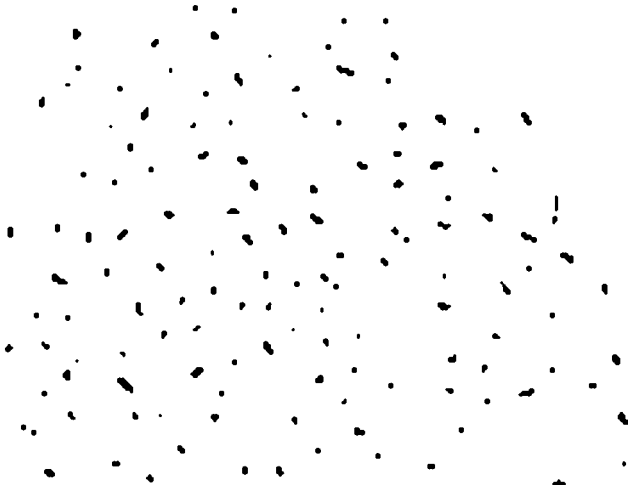


Fig. 54 Dilation of the last erosion.

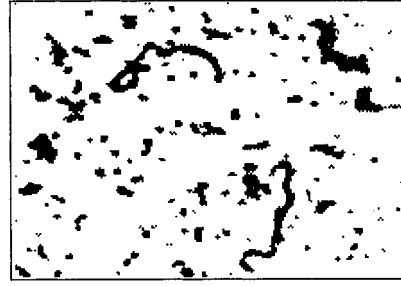


Fig. 55 Threshold.

is called the (morphological) *skeleton of parameter B* (Ref. 4, p. 207).

The operator ϱ_B from $\{0,k\}^E$ to $\{0,k\}^E$, given by

$$\varrho_B = \bigvee \{ \varepsilon_B^i - \gamma_{B, \varepsilon_B^{i+1}} \varepsilon_B^i : i = 0, 1, \dots \}, \quad (80)$$

is called *last erosion of parameter B* (Ref. 1, p. 28).

The image of Figure 54 is the result of the application of this operator on the image of Figure 28. The resulting image has been dilated by the elementary square for better visualization.

The operator $\beta_{n,B}$ from $\{0,k\}^E$ to $\{0,k\}^E$, given by, for any $g \in K^E$,

$$\beta_{n,B}(g) = \bigvee \{ \varepsilon_B^i(g) - \delta_{B,g}^n(\varepsilon_B^{i+1}(g)) : i = 0, 1, \dots \} \quad (81)$$

is called *n-order conditional bisector of parameter B* (Ref. 3, p. 383).

In fact, the conditional bisector is an intermediate operator between the last erosion and the morphological skeleton, depending on the choice of the parameter n it is near to one or other. For $n=0$, the conditional bisector will be the morphological skeleton itself. For values of n bigger than a given minimum, the conditional bisector will be the last erosion itself.

These operators are frequently used to produce markers that identify overlapping objects. The image of Figure 54 is an example of use of the last erosion with this purpose. The objects in this image are markers for the cells of the image of Figure 28.

2.11 Sup-generating and Inf-generating Operators

The sup-generating operators are useful to detect shapes. They act as a template matching where some tolerance is permissible for the matching. This property is achieved by verifying if the shapes to be transformed are limited by two fixed shapes that define the interval that characterizes the operator. A sup-generating operator can also be built in terms of intersection of an erosion and an anti-dilation. Its dual, the inf-generating operator, can be built in terms of

union of an anti-erosion and a dilation. They have these names, because with the supremum of sup-generating operators or the infimum of inf-generating operators we can perform any complete lattice operator.⁶

The *adaptive threshold* of f with respect to f_1 and f_2 , denoted $f_1 \leq f \leq f_2$, is the function in $\{0,k\}^E$ given by, for any $x \in E$,

$$(f_1 \leq f \leq f_2)(x) = \begin{cases} k & \text{if } f_1(x) \leq f(x) \leq f_2(x), \\ 0 & \text{otherwise.} \end{cases} \quad (82)$$

The operator $f_1 \leq \cdot \leq f_2$ from K^E to $\{0,k\}^E$ is called *adaptive threshold* transformation of parameters f_1 and f_2 . Actually, we have

$$f_1 \leq f \leq f_2 = (f_1 \leq f) \wedge (f \leq f_2), \quad (83)$$

where $f_1 \leq \cdot$ is an erosion and $\cdot \leq f_2$ is an anti-dilation.

The *equality* between f_1 and f_2 , denoted $f_1 \equiv f_2$, is the function in $\{0,k\}^E$ given by, for any $x \in E$,

$$(f_1 \equiv f_2)(x) = \begin{cases} k & \text{if } f_1(x) = f_2(x) \\ 0 & \text{otherwise.} \end{cases} \quad (84)$$

The binary operation \equiv from $K^E \times K^E$ to $\{0,k\}^E$ is called *equality* operation. Actually, we have

$$(f_1 \equiv f_2) = f_1 \leq f_2 \leq f_1. \quad (85)$$

The image of Figure 55 is the result of the thresholding (with f_1 and f_2 being two constant functions such that $f_1 \leq f_2$) of the image of Figure 52.

Let F be the set representation for the function $f \in \{0,k\}^E$. Let W be a finite subset of \mathbb{Z}^2 and let A and B be two subsets of W such that $A \subset B$. The *sup-generating* and *inf-generating* of f with respect to (A,B) , denoted $\lambda_{A,B}(f)$ and $\mu_{A,B}(f)$, is the function in $\{0,k\}^E$, given by

$$\lambda_{A,B}(f)(x) = k \Leftrightarrow x \in \{y \in E : F \cap (W+y) \in [A+y, B+y]\}, \quad (86)$$

for any $x \in E$, and

$$\mu_{A,B}(f) = -\lambda_{A,B}(-f). \quad (87)$$

The operators $\lambda_{A,B}$ and $\mu_{A,B}$ are called *sup-generating* and *inf-generating* operators of parameters A and B .^{4,14}

The sup-generating operators are useful to detect bidimensional shapes. For example, the interval of extremities

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

characterizes the sup-generating operator which detects isolated points (according to 8 connectivity), while the interval of extremities

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mathbf{1} & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

detects horizontal segments of size one or two.

A possible equivalent representation for $\lambda_{A,B}$ is, for any $f \in K^E$,

$$\lambda_{A,B}(f) = \varepsilon_A(f) \vee \varepsilon_{B^c}(-f), \tag{88}$$

where B^c is the complement of B relative to W . Note that this representation is an intersection of an erosion of the image with an erosion of its complement. Inspired by this property, Serra (Ref. 3, p. 39) called $\lambda_{A,B}$ *hit-miss* operator of parameters A (hit) and B^c (miss).

As the hit-miss representation of the sup-generating operators is classical, usually we find in the literature the characterization of the sup-generating operators by A and B^c . In the following sections of this paper we also adopt this convention.

Finally, we should say that the hit-miss representation is in fact an intersection of an erosion with an anti-dilation. However, the decomposition of the sup and inf-generating operators in terms of elementary operators can be more easily seen in the following representations:

$$\lambda_{A,B} = \varepsilon_A \wedge -\delta_{B^c} \tag{89}$$

and

$$\mu_{A,B} = \delta_A \vee -\varepsilon_{B^c}. \tag{90}$$

Table 2 presents some intervals that are commonly used as parameter for sup- and inf-generating operators.

The next pair of operators are the *sup-decomposition* (i.e., supremum of sup-generating operators) and the *inf-decomposition* (i.e., infimum of inf-generating operators) general representations for binary operators.

Let \mathcal{A} and \mathcal{B} be two finite sequences of n subsets with elements A_i and B_i , such that $A_i \subset B_i$. The two operators $\psi_{\mathcal{A},\mathcal{B}}$ and $\omega_{\mathcal{A},\mathcal{B}}$ from $\{0,k\}^E$ to $\{0,k\}^E$, given by the following $n-1$ operations

$$\psi_{\mathcal{A},\mathcal{B}} = \vee \{ \lambda_{A_i, B_i} : i = 1, \dots, n \} \tag{91}$$

and

$$\omega_{\mathcal{A},\mathcal{B}} = \wedge \{ \mu_{A_i, B_i} : i = 1, \dots, n \}, \tag{92}$$

are called *n-sup-canonical* operator and *n-inf-canonical* operator of parameters \mathcal{A} and \mathcal{B} .

Let A be a subset of W , i be a positive integer, and $q \in \{0,45,90,180\}$. We denote by $A_{i,q}$ (or simply A_i , when q is fixed in the context) the subset built by an integer rotation by $i \times q$ degrees of A . For example,

$$\text{if } i=2, q=45, \text{ and } B^c = \begin{bmatrix} 0 & 0 & 0 \\ 1 & \mathbf{0} & 1 \\ 1 & 1 & 1 \end{bmatrix},$$

then

$$B_{2,45}^c = \begin{bmatrix} 1 & 1 & 0 \\ 1 & \mathbf{0} & 0 \\ 1 & 1 & 0 \end{bmatrix}.$$

The image of Figure 56 is the result of the application of a sup-decomposition operator that detects end points of digital lines on the image of Figure 57. This operator is characterized by the sequences $A_{i,q}$ and $B_{i,q}^c$, where A and B^c are the end point parameters of Table 2, $q=45$, and $i \in \{0,1,2,\dots,7\}$. The resulting image has been dilated by the elementary square for better visualization.

2.12 Thinning and Thickening

This family of operators is particularly useful for binary image processing, where it is used to build skeletons and detect geometrical properties. These operators are based on two simple operators: thinning and thickening. The thinning operator eliminates the center of shapes detected by the hit-miss operator, while the thickening operator adds the center of the detected shapes.

Let A and B be two subsets of W , such that $A \subset B$. The two operators $\sigma_{A,B}$ and $\tau_{A,B}$ from $\{0,k\}^E$ to $\{0,k\}^E$, given by

$$\sigma_{A,B} = I - \lambda_{A,B} \tag{93}$$

and

$$\tau_{A,B} = I \vee \lambda_{A,B}, \tag{94}$$

are called *thinning* and *thickening* by (A,B) (Ref. 3, p. 390).

Let g be an element of $\{0,k\}^E$. The operators $\sigma_{A,B,g}$ and $\tau_{A,B,g}$ from $\{0,k\}^E$ to $\{0,k\}^E$, given by

$$\sigma_{A,B,g} = \sigma_{A,B} \vee g \tag{95}$$

and

$$\tau_{A,B,g} = \tau_{A,B} \wedge g, \tag{96}$$

are called *conditional thinning* and *thickening* by (A,B) given g (Ref. 3, p. 393).

Let \mathcal{A} and \mathcal{B} be two infinite sequences of subsets of W , respectively, with elements A_i and B_i such that $A_i \subset B_i$.

Table 2 Some useful pairs of structural elements for canonical operators, thinning, and thickening.

Struct. Element A	Struct. Element B ^c	[A, B]	Canonical operator	Thinning	Thickening
0 0 0	1 1 1	0 0 0		Homotopic skeleton	-
0 1 0	0 0 0	x 1 x	-		
1 1 1	0 0 0	1 1 1			
0 0 0	1 1 1	0 0 0			Homotopic Exoskeleton
0 0 0	0 0 0	x x x	-	-	
1 1 1	0 0 0	1 1 1			
0 0 0	1 0 0	0 x x		Homotopic marking	-
0 1 1	1 0 0	0 1 1	-		
0 0 0	1 0 0	0 x x			
0 0 0	0 0 0	x x x		Skeleton	
0 1 0	1 0 1	0 1 0	End Points	Pruning	-
0 0 0	1 1 1	0 0 0			
0 0 0	0 0 0	x x x	Background		Exoskeleton
1 0 1	0 1 0	1 0 1	End Points	-	Pruning
1 1 1	0 0 0	1 1 1			
0 0 0	1 1 1	0 0 0		Point Cleaning	
0 1 0	1 0 1	0 1 0	Isolated Points		-
0 0 0	1 1 1	0 0 0			
1 1 1	0 0 0	1 1 1			Pointwise Hole
1 0 1	0 1 0	1 0 1	Pointwise Hole	-	Cleaning
1 1 1	0 0 0	1 1 1			
1 0 1	0 1 0	1 0 1			
0 1 0	1 0 1	0 1 0	Triple Points	-	-
0 1 0	0 0 0	x 1 x			
1 0 1	0 1 0	1 0 1			
0 1 0	1 0 1	0 1 0	Triple Points	-	-
1 0 0	0 0 1	1 x 0			
1 0 1	0 1 0	1 0 1			
0 1 0	1 0 1	0 1 0	Triple Points	-	-
0 0 1	1 0 0	0 x 1			
1 1 1	0 0 0	1 1 1		Boundary	-
1 1 1	0 0 0	1 1 1	-		
1 1 1	0 0 0	1 1 1			

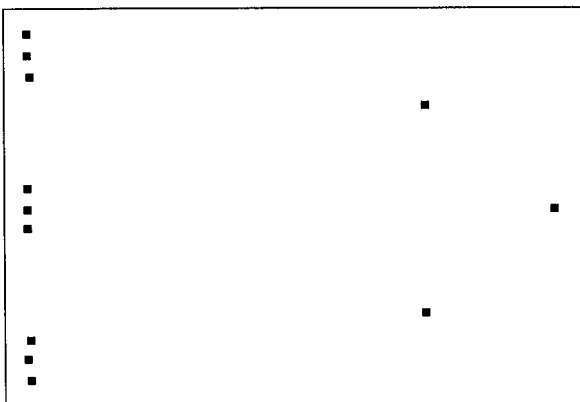


Fig. 56 Dilated end points.

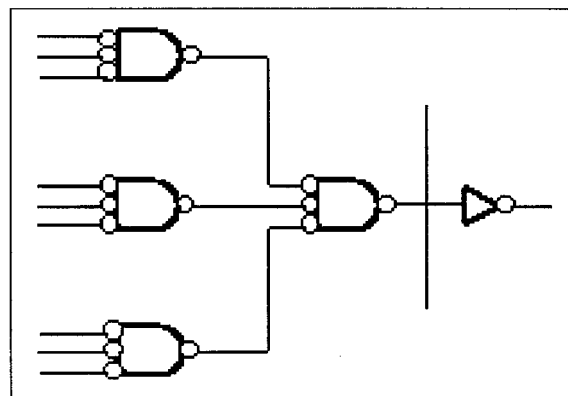


Fig. 57 Logic circuit.

The two operators $\sigma_{n,\mathcal{A},\mathcal{B}}$ and $\tau_{n,\mathcal{A},\mathcal{B}}$ from $\{0,k\}^E$ to $\{0,k\}^E$, given by the following $n-1$ successive compositions (using the first n elements of \mathcal{A} and \mathcal{B})

$$\sigma_{n,\mathcal{A},\mathcal{B}} = \sigma_{A_n,B_n} \cdots \sigma_{A_1,B_1} \tag{97}$$

and

$$\tau_{n,\mathcal{A},\mathcal{B}} = \tau_{A_n,B_n} \cdots \tau_{A_1,B_1}, \tag{98}$$

are called n -thinning and n -thickening of parameters \mathcal{A} and \mathcal{B} (Ref. 3, p. 390).

Figure 3 presents the result of an application of the n -thinning operator on the image of Figure 2. This operator performs the successive elimination of end points of digital line segments and is characterized by a sequence built by repetitions of the sequence of structural elements used in the operator that detects end points.

The two operators $\Sigma_{\mathcal{A},\mathcal{B}}$ and $T_{\mathcal{A},\mathcal{B}}$ from $\{0,k\}^E$ to $\{0,k\}^E$, given by

$$\Sigma_{\mathcal{A},\mathcal{B}} = \sigma_{\infty,\mathcal{A},\mathcal{B}} \tag{99}$$

and

$$T_{\mathcal{A},\mathcal{B}} = \tau_{\infty,\mathcal{A},\mathcal{B}}, \tag{100}$$

are called *skeleton by thinning* and *exoskeleton by thickening* of parameters \mathcal{A} and \mathcal{B} (Ref. 4, p. 202).

Figure 2 presents the result of the application of the skeleton by thinning operator on the image of Figure 1. This operator is characterized by the sequences A_{qi} and B_{qi} , where $q=45$, and A and B^c are the parameters for the homotopic skeleton given in Table 2. A remarkable property of this skeleton is that it preserves the topological structure of the original image.

Let g be an element of $\{0,k\}^E$. The two operators $\sigma_{n,\mathcal{A},\mathcal{B},g}$ and $\tau_{n,\mathcal{A},\mathcal{B},g}$ from $\{0,k\}^E$ to $\{0,k\}^E$, given by the following $n-1$ successive compositions

$$\sigma_{n,\mathcal{A},\mathcal{B},g} = \sigma_{A_n,B_n,g} \cdots \sigma_{A_1,B_1,g} \tag{101}$$

and

$$\tau_{n,\mathcal{A},\mathcal{B},g} = \tau_{A_n,B_n,g} \cdots \tau_{A_1,B_1,g}, \tag{102}$$

are called n -conditional thinning and n -conditional thickening of parameters \mathcal{A} and \mathcal{B} given g .

The operators $\Sigma_{\mathcal{A},\mathcal{B},g}$ and $T_{\mathcal{A},\mathcal{B},g}$ from $\{0,k\}^E$ to $\{0,k\}^E$, given by the following successive infinite compositions

$$\Sigma_{\mathcal{A},\mathcal{B},g} = \sigma_{\infty,\mathcal{A},\mathcal{B},g} \tag{103}$$

and

$$T_{\mathcal{A},\mathcal{B},g} = \tau_{\infty,\mathcal{A},\mathcal{B},g}, \tag{104}$$

are called *conditional skeleton by thinning* and *conditional exoskeleton by thickening* of parameters \mathcal{A} and \mathcal{B} given g .

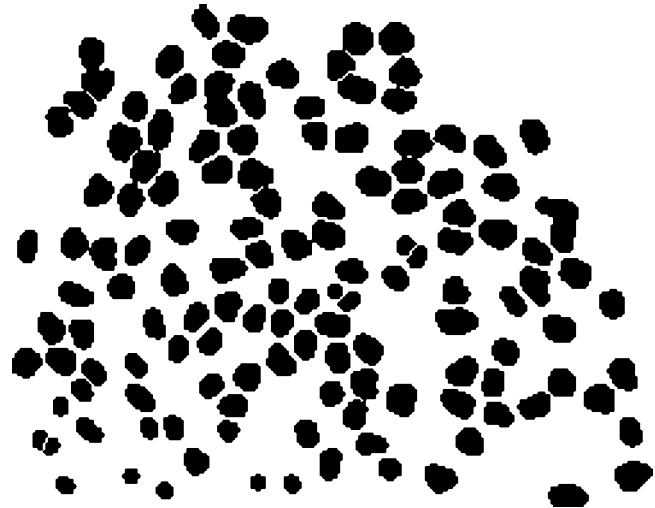


Fig. 58 Conditional exoskeleton by thickening.

Let us denote the infinite sequence A_{iq} simply by A_q . Figure 58 presents the result of the application of the conditional exoskeleton by thickening on the image of Figure 54, conditioned by the image of Figure 28. This operator is characterized by the sequences \mathcal{A}_q and \mathcal{B}_q , where $q=45$, and A and B are the parameters for the homotopic exoskeleton given in Table 2.

Note that the effect of this operator is similar to the inf-reconstruction operator, but it preserves the number of objects of the marker.

Let A and B^c and C and D^c be the parameters for the homotopic skeleton and for the detection of end points given in Table 2. The operator Σ from $\{0,k\}^E$ to $\{0,k\}^E$, given by, for any $f \in K^E$,

$$\Sigma(f) = \Sigma_{\mathcal{C}_{45},\mathcal{D}_{45}}[\Sigma_{\mathcal{A}_{45},\mathcal{B}_{45}}(-f)], \tag{105}$$

is called *skiz* (Ref. 1, p. 260.)

The first operator is a skeleton that preserves the topology of the complement of the input image, while the second one is a pruning of this skeleton. The *skiz* gives a partition of the domain E . Figure 59 presents an application of the *skiz* operator on the image of Figure 60.

Let C and D^c be the parameters for the homotopic thickening given in Table 2 and let $g, f \in \{0,k\}^E$, such that $\varepsilon_{nB}(f) \leq g$. The operator $T_{n,g}$ from $\{0,k\}^E$ to $\{0,k\}^E$ given by

$$T_{n,g}(f) = T_{\mathcal{C}_{45},\mathcal{D}_{45},f}(T_{\mathcal{C}_{45},\mathcal{D}_{45},\varepsilon_{45}f}) \cdots (T_{\mathcal{C}_{45},\mathcal{D}_{45},\varepsilon_{(n-2)}f}) \times (T_{\mathcal{C}_{45},\mathcal{D}_{45},\varepsilon_{(n-1)B}}(f)(g) \vee g) \vee g \vee g, \tag{106}$$

is called n -smoothing segmentation (Ref. 4, p. 205).

The effect of the smooth segmentation is similar to the one of the conditional exoskeleton by thickening, that is, it partitions the input image keeping the topology of the marker. However, usually the smooth segmentation pro-

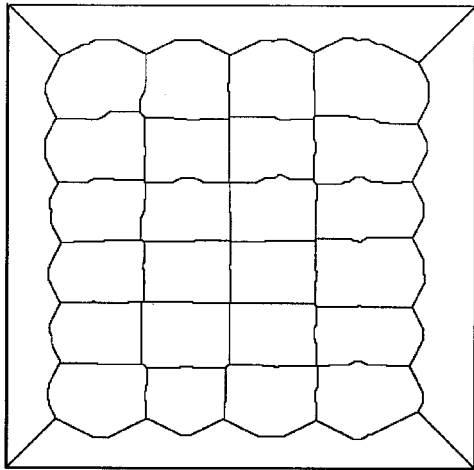


Fig. 59 Skiz.

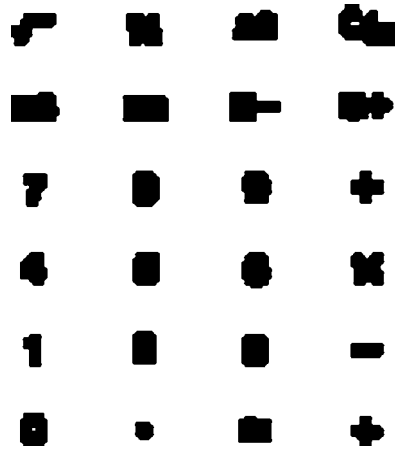


Fig. 60 Internal markers for the keys.

duces division lines better positioned. This happens because the smooth segmentation does successive reconstructions of n -erosions, while the exoskeleton does a direct reconstruction.

The operator Ω from K^E to $\{0,k\}^E$, given, for any $f \in K^E$, by

$$\Omega(f) = T_{\mathcal{C}_{45}, \mathcal{D}_{45}, f \leq k} (T_{\mathcal{C}_{45}, \mathcal{D}_{45}, f \leq k-1} \dots (T_{\mathcal{C}_{45}, \mathcal{D}_{45}, f \leq 2} \times (T_{\mathcal{C}_{45}, \mathcal{D}_{45}, f \leq 1} e_B^{\min}(f)) \vee e_B^{\min}(f)) \dots \vee e_B^{\min}(f)) \vee e_B^{\min}(f) \quad (107)$$

is called *watershed*.¹⁵

If we interpret f as a topographic surface, then the watershed produces a partition of the domain E , where each part is the domain of a catchment basin. Figure 61 shows the catchment basins separated by dams in the graph of a 1-D function.

To get the catchment basins the operator performs successive reconstructions of cuts at consecutive levels: the cut

at level one is reconstructed from the regional minima, the cut at level two is reconstructed from union of the result of the previous reconstruction and the regional minima and so on.

$$\text{Let } A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mathbf{1} & \mathbf{1} \\ 0 & 1 & 0 \end{bmatrix}, \quad B^c = \begin{bmatrix} 1 & 1 & 0 \\ 1 & \mathbf{0} & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 0 & 0 \\ 1 & \mathbf{1} & \mathbf{1} \\ 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad D^c = \begin{bmatrix} 0 & 1 & 0 \\ 0 & \mathbf{0} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

and let \mathcal{A}_{45} , \mathcal{B}_{45} , \mathcal{C}_{45} , and \mathcal{D}_{45} be the respective infinite sequences created by rotations of these four subsets. The operator $\Sigma_{\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}}$ from $\{0,k\}^E$ to $\{0,k\}^E$, given by the following succession of compositions

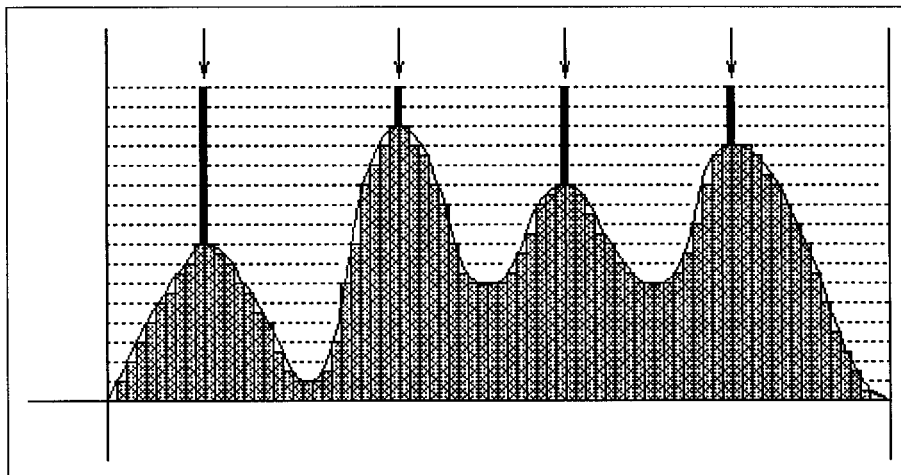


Fig. 61 Watershed.

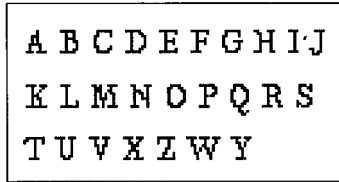


Fig. 62 Skeleton by thinning.

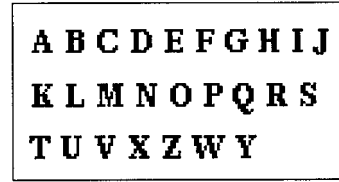


Fig. 64 Letters.

$$\Sigma_{\mathcal{A}_{45}, \mathcal{B}_{45}, \mathcal{C}_{45}, \mathcal{D}_{45}} = \dots (\sigma_{A_i, B_i} \wedge \sigma_{A_{i+1}, B_{i+1}} \wedge \sigma_{C_i, D_i}) \dots (\sigma_{A_1, B_1} \wedge \sigma_{A_2, B_2} \wedge \sigma_{C_1, D_1}) \quad (108)$$

is called *refined skeleton by thinning*.¹⁶

Figures 62 and 63 show, respectively, the images produced by the application of the skeleton by thinning, using the homotopic parameters, and its refinement to the image of Figure 64.

3 Examples of Applications

For demonstrating the potential of MMach, we present in this section some examples of applications of these tools to image analysis. These applications illustrate the use of image analysis in cytology, industrial automation, and medicine.² (Note that these and other examples are deposited in the subdirectory workspaces of the MMach tool-box.)

3.1 Noisy Edge Detection

The goal in this example is to detect edges of objects on images corrupted by noise. The original image f_1 (Figure 38) is a gray-scale image that presents a disk corrupted by additive Gaussian noise. By using an alternated sequential filter with three stages and two structural elements B_1 and B_2 , we get the filtered image f_2 (Figure 37):

$$f_2 = \gamma_{3B_2} \phi_{3B_2} \gamma_{3B_1} \phi_{3B_1} \gamma_{2B_2} \phi_{2B_2} \gamma_{2B_1} \times \phi_{2B_1} \gamma_{B_2} \phi_{B_2} \gamma_{B_1} \phi_{B_1}(f_1), \quad (109)$$

where B_1 is the elementary cross and B_2 is the elementary square.

By using a threshold transformation with threshold value 20 and an internal contour extractor, we get the contour image f_4 (Figure 65):

$$f_4 = f_3 - \varepsilon_{B_1}(f_3), \quad (110)$$

where

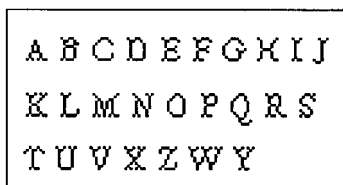


Fig. 63 Refined skeleton by thinning.

$$f_3 = (20 \leq f_2) \quad (111)$$

and 20 means a constant image.

3.2 Segmentation of Overlapped Objects

The goal in this example is to separate overlapped (binary) objects, for instance, to count the number of objects in an image. This procedure may be useful in cytology or in industrial automation. The solution for this problem constitutes an interesting application of the distance function.

The original image f_1 (Figure 28) presents some overlapped blobs. This image is binary so we apply the distance function operator to get a landscape model of the image. Image f_2 (Figure 29) presents the result of the distance function:

$$f_2 = \Psi_d(f_1). \quad (112)$$

Image f_3 is the result of the regional maximum operator, followed by a dilation (large enough to ensure that there is only one marker for each blob) and an erosion (to ensure that they will not cause an ill positioning of the watershed lines):

$$f_3 = \varepsilon_B^2(\delta_B^2(\mathcal{Q}_B^{\max}(f_2))), \quad (113)$$

where B is the elementary square. Figure 49 presents a composition of images f_1 and f_3 , which permits a better visualization of the position of the markers.

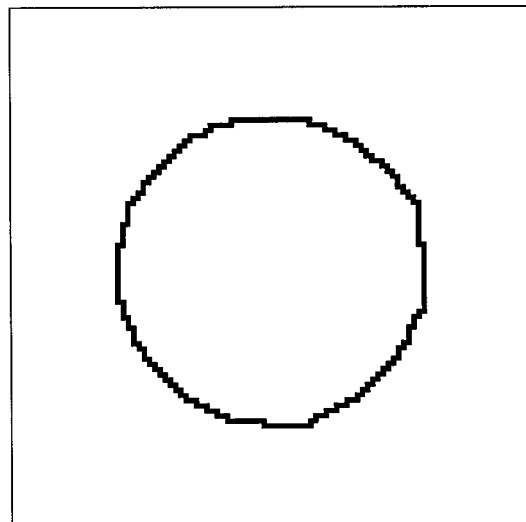


Fig. 65 Edges.

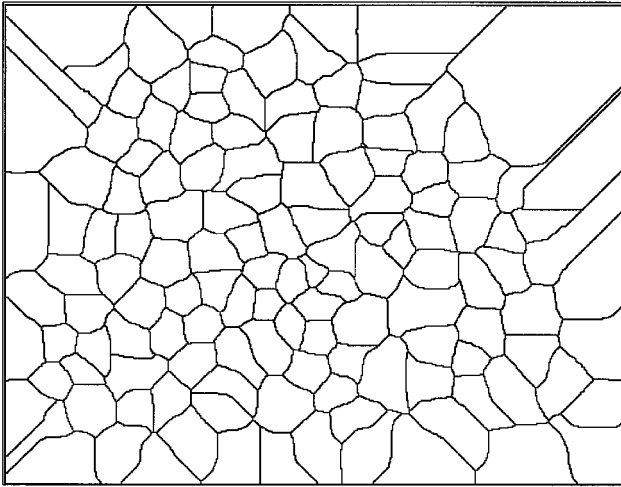


Fig. 66 Watershed lines.

Image f_4 (Figure 66) is the negation of the image obtained by the application of the watershed operator to the image created by the negation of the union of the marker image and the distance function image:

$$f_4 = -\Omega[-(f_2 \vee f_3)]. \quad (114)$$

Image f_5 (Figure 67) is obtained by the subtraction of images f_1 and f_4 :

$$f_5 = f_1 - f_4. \quad (115)$$

As the borders are not very smooth, we should apply a morphological filter. Image f_6 (Figure 68) is the result of an opening by a disk:

$$f_6 = \gamma_B(f_5), \quad (116)$$

where B is an Euclidean disc of diameter 7.

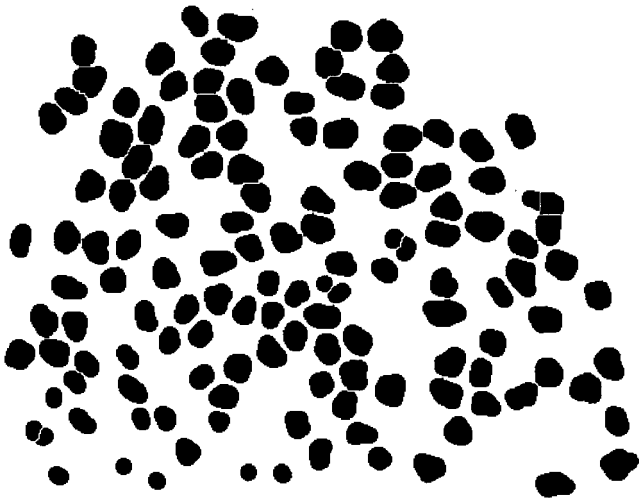


Fig. 67 Subtraction.

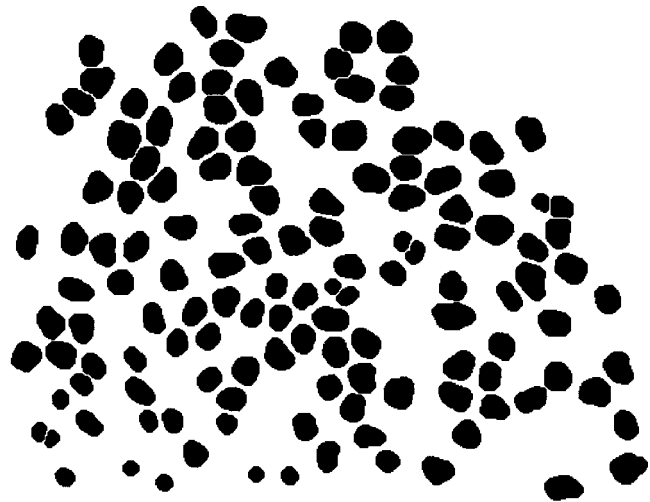


Fig. 68 Opening.

3.3 Automatic Quality Control on Printed Circuit Boards

The goal in this example is to detect holes on printed circuit board (PCB) images. This may be useful in quality control, for instance, if a board has less holes than were expected, then this board may be rejected.

The original image f_1 (Figure 69) presents a circuit board from where we want to extract the holes. The holes are presented in image f_2 (Figure 70), which is obtained by applying the operator close holes and subtracting the original image from that image:

$$f_2 = \Phi_B(f_1) - f_1, \quad (117)$$

where B is the elementary square.

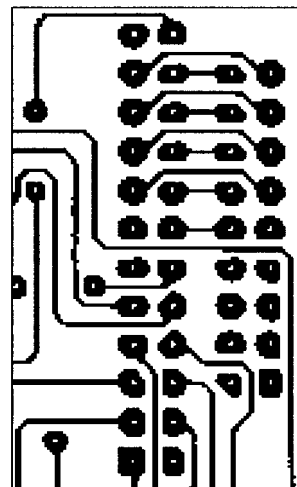


Fig. 69 PCB.

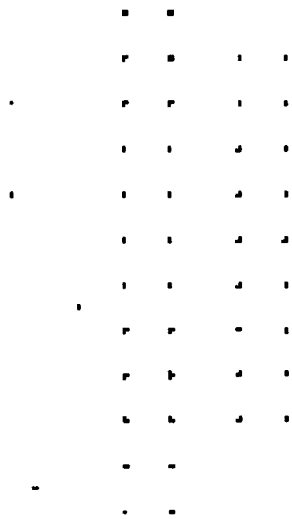


Fig. 70 Holes of the PCB.

3.4 Keys of a Pocket Calculator Machine

The goal in this example is to identify objects in the image of a pocket calculator. Two segmentation problems can be explored in that image: extraction of the digits printed on the keys and of the keys themselves.

The original image f_1 (Figure 51) presents part of the panel of a pocket calculator machine. In this example let B be the elementary square.

3.4.1 Digits segmentation

Image f_2 (Figure 71) presents an enhancement of the digits that is obtained by an opening by reconstruction top hat transformation:

$$f_2 = \Gamma_{A,B}^t(f_1), \tag{118}$$

where $A = 5B$. The image has been inverted for better visualization.

Image f_3 (Figure 72) shows the result of a threshold

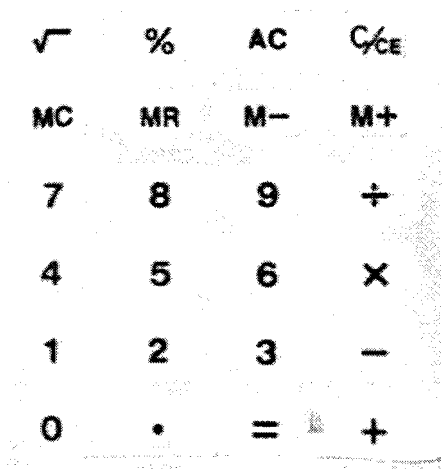


Fig. 71 Top Hat.

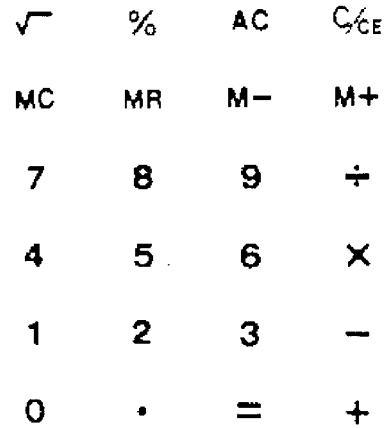


Fig. 72 Result of the digits segmentation.

applied to the top hat:

$$f_3 = (128 \leq f_2 \leq 255), \tag{119}$$

where 128 and 255 represent two constant images.

3.4.2 Edges segmentation

The successive composition of the morphological gradient, the imposition of minima and the watershed constitutes a very powerful tool for image segmentation that is known as Beuchers' paradigm.¹⁵ The division lines given by the watershed operator when applied on the gradient filtered by the minima imposition will give exactly the edges of the objects pointed by the marker. The main quality of this method is changing the problem of heuristic edge detection to the problem of heuristic detection of markers to objects, which is a much simpler problem. Let us use this approach to detect the keys of the calculator.

A good internal marker for a key is the digit printed on it, but as a marker it must be a connected component. It is necessary to apply some transformation to image f_3 . Image f_4 (Figure 60) is the result of an n -dilation applied to f_3 :

$$f_4 = \delta_B^n(f_3). \tag{120}$$

The external markers are obtained by applying the Skiz operator on the image f_4 (with an extended domain). Image f_5 (Figure 73) is the result of the union of the internal and external markers:

$$f_5 = f_4 \vee \Sigma(f_4). \tag{121}$$

By applying the morphological gradient operator on the original image we get image f_6 :

$$f_6 = \Psi_{B,B}(f_1). \tag{122}$$

Figure 50 shows the inverse of image f_7 for better visualization of the details.

Beucher's paradigm says that we have to impose new minima to the gradient image according to the markers found. Image f_7 (Figure 74) is the result of this procedure,

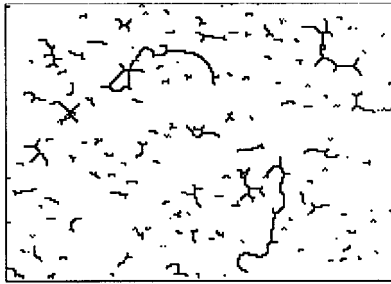


Fig. 76 Skeleton.

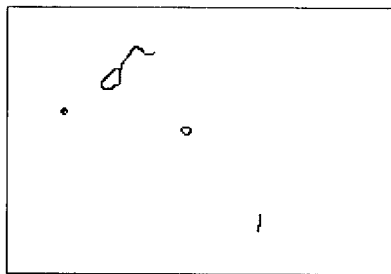


Fig. 77 First pruning.

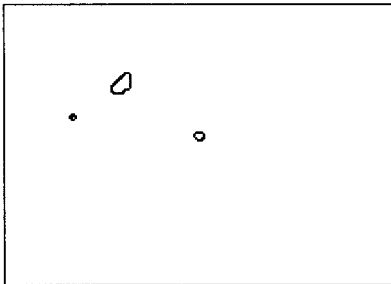


Fig. 78 Second pruning.

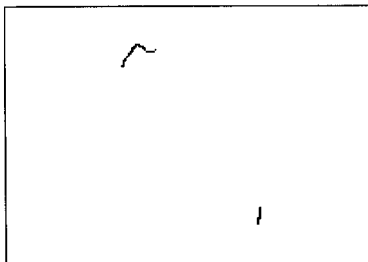


Fig. 79 Residue.



Fig. 80 Worms.

where \mathcal{E}_{45} and \mathcal{D}_{45} are built from the parameters for the skeleton pruning given in Table 2.

The remaining lines are part of the worms so they can be used as markers to reconstruct them, but first it is necessary to separate them from the rounded objects. Image f_7 (Figure 78) presents the rounded objects that are obtained by applying once more a pruning of the skeleton:

$$f_7 = \Sigma_{\mathcal{E}_{45}, \mathcal{D}_{45}}(f_6). \quad (130)$$

Image f_8 (Figure 79) presents the markers for the worms that are the result of the subtraction of image f_7 from f_6 :

$$f_8 = f_6 - f_7. \quad (131)$$

Finally, image f_9 (Figure 80) presents the filariosis transmitters, which are obtained by the inf-reconstruction of image f_4 from image f_8 :

$$f_9 = \gamma_{B, f_8}(f_4). \quad (132)$$

Figure 81 shows the composition of the segmented image with the original image.

4 KHOROS System

KHOROS^{17,18} is a software environment designed for research on image processing. It has been created at the Department of Electrical and Computer Engineering of the University of New Mexico, Albuquerque, and has become very popular. According to recent statistics of the KHOROS group, it has nearly 10,000 users around the world that can have support and exchange information through a very active mailing list.

Since image processing encompasses a wide spectrum of applications, it has been designed from a very broad per-

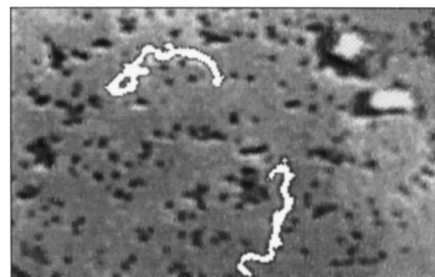


Fig. 81 Worms in the original image.

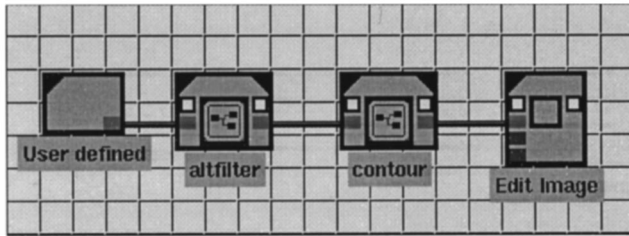


Fig. 82 Workspace.

spective. For example, it includes mechanisms for distributed computing, interactive visualization of many data types, and suitable user interfaces.

One of the most powerful features of KHOROS is CANTATA, its high-level abstract interface. CANTATA is a graphically expressed, data-flow oriented language that provides a visual programming environment for the system. Data flow is an approach in which a program is described as a directed graph, where each node represents an operation (or function) and each direct arc represent a path over which data tokens flow. A CANTATA program is called workspace. Figures 82, 83, and 84 are examples of workspaces.

KHOROS has been designed to be portable and extensible. It relies on existing standards (X Windows and UNIX), incorporates tools for software development and maintenance (a high level user interface specification and a code generation tool set), a flexible data exchange format, tools to export and import standard data formats, and an algorithm library.

The user programs can be organized as independent sub-systems, called toolboxes, that can easily be integrated to the system. Usually, a new toolbox is deposited at ftp.khoros.unm.edu and can be accessed by the community of KHOROS users via anonymous ftp.

KHOROS has been extensively used¹⁹⁻²² to perform image processing research, algorithm development, and data

visualization. In fact, the known applications cover a very broad spectrum: industrial inspection, medical diagnosis, optical measurement, remote sensing, semiconductor processing, optics, medical imaging, ecosystem analysis, cell biology, etc.

5 Mathematical Morphology Toolbox for KHOROS

We implemented MMach (a contraction for a morphological machine) for 1-D and 2-D gray-scale or binary images as a KHOROS toolbox, where each family of morphological operators is presented as a submenu of the toolbox main menu.

5.1 Architecture

Following the theory of mathematical morphology, most of the operators are built by composition of the elementary operators and operations.

The dilations and erosions can be further decomposed in terms of dilations and erosions by subsets of the elementary square, but there are also available algorithms for erosion and dilation by a generic structural element. Yet in the case of gray-scale image processing, the generic structural element may be flat or not.

The structural elements are represented by a special data structure. This structure is more general than the one used to represent images, since the structural element domain is not necessarily rectangular. So, of course, images may be converted to this structure, but the converse is not true.

As the elementary operators for binary images have some additional properties over the corresponding ones for gray-scale images, different algorithms were chosen in each case. Special algorithms for distance function, inf and sup-reconstruction, labeling, and watershed are also available.

To simplify its use, the system has been designed to be data type oriented, that is, when executing a given operation or operator that makes sense on different data types, it switches automatically to the most efficient algorithm for

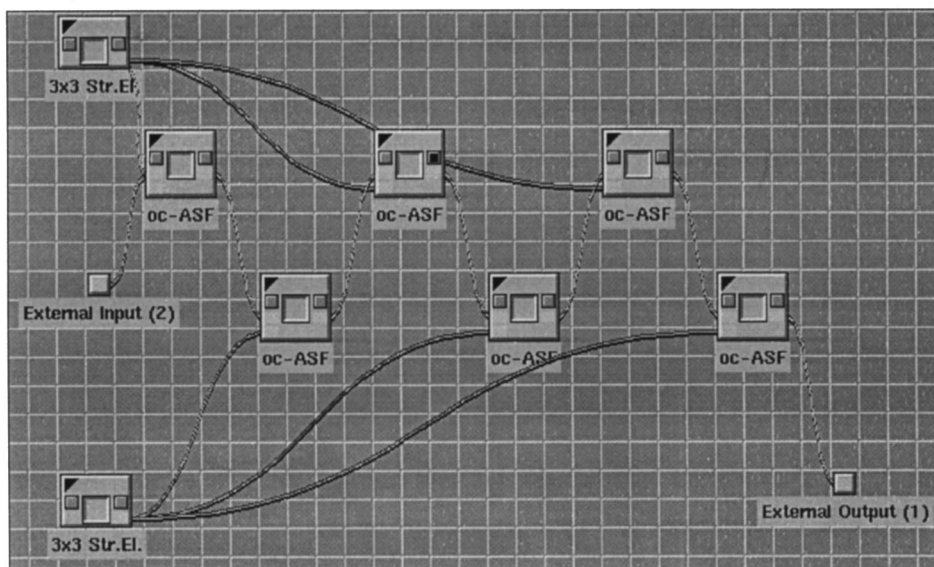


Fig. 83 Subroutine that performs the ASF filter (altfilter).

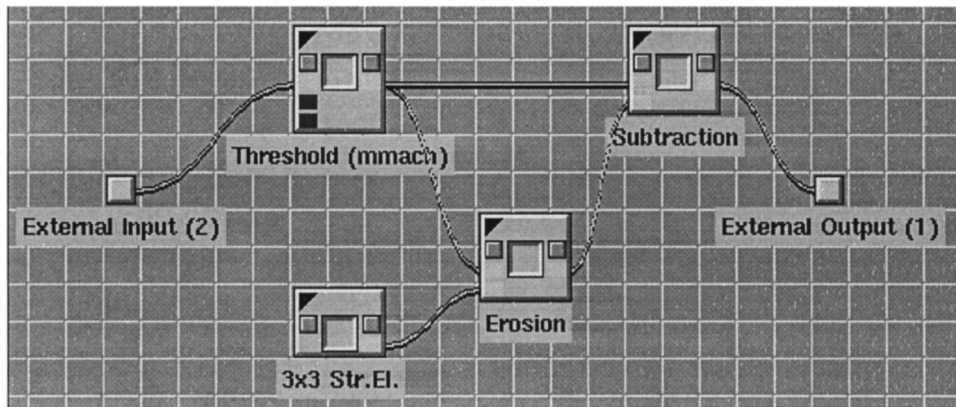


Fig. 84 Subroutine that performs the contour extraction (contour).

the current input data. This polymorphism property is quite important, since it permits to represent the operators with all their theoretical generality keeping the performance of its use.

We should also point out that all the operators built from dilations and erosions (characterized by structural elements) are polymorphic. However, we present just the binary definition for some of them, as some sup- and inf-generatings, some thinnings and thickenings, and some residues. We have adopted this presentation because the behavior of these operators on gray-scale images is not well known.

Complex operators can be built either as CANTATA or C programs, using main programs or subroutines of the available primitives.

Figure 82 presents the workspace constructed to solve the application example 1 (detection of edges on noisy images). This workspace is composed of four icons: user defined, which gets the original image; altfilter, which performs the alternated sequential filtering; contour, which performs the edge extraction; and edit image, which opens a visualization window. Actually, the icons altfilter and contour are data-flow subroutines presented in Figures 83 and 84.

5.2 Contents

The toolbox is composed by several groups of programs: relations, operations, structural elements, dilations and erosions, morphological filters, connected filters, residues, sup-generating and inf-generating operators, and thinning and thickening (Table 1).

The right choice of the parameters for these operators gives a large number of tools to extract image information: image sharpening and smoothing, threshold segmentation, elimination of particles that hit the image edges, closing of holes, size distributions, skeletons and their characteristics points (triple, end, etc.), geometrical segmentation and filtering, etc. For example, Table 2, which has been adapted from Ref. 3, p. 392 for the square grid, gives some useful pairs of structural elements for operators built from the sup- and inf-generating operators.

MMach has some tools for the creation of structural elements: small structural elements may be edited interactively; disks, semispheres, and cylinders may be defined

parametrically and images may be converted into structural elements. A complete specification of a structural element involves the definition of its origin, so we have attended for this point: in a disk the origin is its center; in a sphere and in a cylinder the origin is the center of its domain; and in an image the origin is the point $(\lfloor n_1/2 \rfloor, \lfloor n_2/2 \rfloor)$, where n_1 and n_2 are the sizes of the image in the horizontal and vertical directions.

The composition of the skeleton by thinning built from the pair of structural elements

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mathbf{1} & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad B^c = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

with the skeleton by thinning built from

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & \mathbf{1} & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad B^c = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

gives a new interesting skeleton.¹⁶

For each program of the toolbox there is an online help associated, which gives the definition of the operators and a set of well-known parameters to extract useful image information.

5.3 Algorithms for Erosions and Dilations Defined by Structural Elements

Since the erosions and dilations defined by structural elements are an important part of the kernel of the system, considerable effort has been put on making them as fast as possible in current general purpose hardware. In order to achieve a better performance, different algorithms were chosen for binary and gray-scale images.

5.3.1 Dilation and erosion for binary images

The key factor used to implement fast erosion and dilation algorithms (by structural elements) for binary images is the inherent parallelism of the 32-bit bitwise operations, found in general purpose CPU instructions set.

As a particular consequence of Property 1, the implementation of binary dilation and erosion can be made by translating the image by all the points of the structural element and taking, respectively, the logical OR and AND of the translated images. By using this formulation, the parallelism can be easily achieved.

To use the intrinsic parallelism of the logical 32-bit bitwise AND and OR operations, we pack the binary images in sets of 32 pixels in a 32-bit integer.

As the packed image is stored in a row by row basis, vertical translations are efficiently handled by adding the current packed pixel address by the width of the packed image. Horizontal translations are implemented by shifting and masking operations to shift 32-bit integer boundaries.

Using this approach we can compute 32 bits in parallel with the additional benefit of reading and writing the image data in a more efficient packed binary pixel format.

In order to achieve more (added) performance, for operators defined by structural elements that are subsets of the elementary square, the image is subdivided in nine image regions: one middle, four corners, and four side regions. Each region is processed separately to avoid unnecessary testing for image boundaries.

The KHOROS has a bit format that already supports a packed image format. The implementation follows the 32-bit parallel algorithm described using structural elements that are subsets of the elementary square. Some optimizations have been made for the limit cases, when the structural element is an empty set or the complete elementary square.

5.3.2 Dilation and erosion for gray-scale images

Following the definition, the implementation of dilation and erosion (by structural elements) can be made by translating the structural element over the input image and taking the local maximum and minimum.

By this approach, the neighborhood of each pixel needs to be accessed; that is, it is necessary to access $n \times N \times M$ array elements, where n is the cardinality of B , and N and M are the number of rows and columns of the image.

Taking as structural element just subsets of the elementary square, this algorithm leads to good implementations for gray-scale dilation and erosion.

The implementation is divided into ten cases, according to the cardinality of the structural element, from zero (empty set) to nine (the complete elementary square). In each case, the structural element points (i.e., the values for local translations) are stored in a corresponding number of fast registers. The nested conditional expressions were open to avoid unnecessary steps.

To achieve a better performance, as in the binary case, the image was subdivided in nine subregions and some optimizations were made for the limit cases.

5.3.3 Performance evaluation

Table 3 shows the performance evaluation for some dilations and erosions in the binary and gray-scale cases.

The execution time for each operator, given in milliseconds (ms), has been calculated from measure of the time spent by a sequence of a thousand operators. The machine

we used was a SUN SPARCstation-2 and the input data were images of size $256 \times 256 \times 1$ (binary) and $256 \times 256 \times 8$ (gray scale).

The speed-up of performing a dilation or an erosion of a binary image by the dedicated algorithm is between 8 to 10 times (Table 3).

5.4 Algorithms for Other Important Operators

In order to achieve a better performance, special algorithms were implemented for commonly used morphological operators. The operators implemented through special algorithms are: distance function, inf- and sup-reconstruction, labeling, and watershed.

5.4.1 Distance function

The distance function algorithms implemented were the ones proposed by Soille (Ref. 23, p. 25). They are based on the following principles: identification of the pixels in the edges of the image; and propagation of the computed distance through a first-in, first-out queue.

When the notion of distance used is the Euclidean, two distance images are computed, one for the horizontal projection and other for the vertical projection. From these two distance projection images the Euclidean distance is computed.

The complexity of these algorithms depends on the number of 1 pixels in the input binary images, when the number of 1 pixels increases the complexity of the algorithm increases. The time spent by the algorithm for the Euclidean distance is almost twice the time spent by the city block and chessboard distances. For a square of side 256, the time spent for computing the city block or chessboard distance function of a 1 constant image through this algorithm was about 2 s on a SPARC station IPX.

5.4.2 Inf- and sup-reconstruction

The reconstruction algorithms implemented for gray-scale images were the ones proposed by Vincent.¹³ These algorithms have a hybrid nature, because they are based on both a sequential or recursive algorithm and an algorithm that uses a queue of pixels.

The recursive algorithm is based on the following principles: the image pixels are scanned in a predefined order,

Table 3 Performance of dilations and erosions.

Structural Element	Binary Image (ms)	Gray-scale Image (ms)	Speed-up
1 1 1			
1 1 1	8.5	71.2	8.4
1 1 1			
0 1 0			
1 1 1	9.1	90.0	9.9
0 1 0			
0 0 0			
1 1 1	5.5	48.0	8.7
0 0 0			

generally raster or antiraster; and the new value of the current pixel, determined from the value of the pixel in the neighborhood, is written directly in the same image, so that it is taken into account when determining the new values of the as yet unconsidered pixels.

The algorithms which use a queue of pixels are based on the following principles:

- consider only the pixels which may be modified
- initializes by a first scanning that detects the pixels that may be modified
- information is propagated only in the relevant image parts.

The hybrid algorithm initiates with two scans of the recursive algorithm, then the algorithm that uses a queue of pixels is applied on the resulting image. The complexity of this algorithm depends strongly on the input images, but experimental results show it has been up to 15 times faster than the algorithm built just from the gray-scale elementary operators and operations.

5.4.3 Labeling

The labeling transforms a binary BYTE image into a gray-scale SHORT image, where each pixel of same connected component of the binary image has the same value in the gray-scale image. The algorithm implemented was the labeling algorithm proposed by Serra (Ref. 3, p. 405); that is, the iteration (from the original binary image until this image is empty) of the following steps:

- identification of the first point of the input image (counting from left to right and from top to bottom)
- inf-reconstruction of the input image from the first point image
- labeling of the identified connected component
- subtraction of the identified connected component from the input image.

The inf-reconstruction step was implemented based on the representation of the objects by their contours and the dilation algorithm for this data structure.²⁴ The subtraction step has been eliminated by writing, during the reconstruction of the connected component, directly in the input image the label of the pixel.

The performance of this algorithm depends mainly on the number of 1 pixels in the input image, when this number decreases the algorithm performance increases. For a square E of side 256, the time for labeling the one constant image through this algorithm was about 2 s on a SPARC station IPX.

5.4.4 Watershed

The watershed algorithm implemented was the one proposed by Soille and Vincent.²⁵ This algorithm simulates the progressive flooding of the picture and is based on the following principles: pixels are sorted in the increasing order of their gray values; and successive gray levels are processed in order to simulate the flooding propagation.

A distributive sort technique combined with a breadth-first scanning of each gray level allows an extremely fast

performance of this algorithm. For example, the computation of the watershed of a 256×256 gray-scale image by this algorithm was about 4 s on a SPARC station IPX.

6 Comparison with Other Freely Available Software for Mathematical Morphology

Based on the "Mathematical Morphology Digest," maintained since 1993 by H. Heijmans at web site <http://www.cwi.nl/projects/morphology/>, there is only one other freely available software for mathematical morphology, by R. A. Peters II, available at <ftp:image.vuse.vanderbilt.edu/pub/morph.tar.Z> since 1993. The major advantages of our software are: operator polymorphism, consistent definition of dilation and erosion, and compatibility with Khoros software platform. We use extensively the concept of operator polymorphism. It means, for instance, we have just a single operator called dilation that deals with binary and gray-level images (8 and 16 bit) of one and two dimensions, using flat and nonflat structuring functions. This simplifies the user interface to the software by reducing the number of available functions and by increasing the flexibility and power of the operators without losing its overall consistency. Most of the time, a solution for a binary image problem can also be used as a solution for gray-scale images without changing the name of the operators being used. The binary general case of erosion in Peters' software is a hit-miss transform, which can be confusing for the mathematical morphology user.

The completeness of the set of operators and the speed enhancements of MMach such as the 32 bit processing for binary images and queue based algorithms for labeling, distance transform, watershed, and image reconstruction are features only available in high quality commercial softwares.

7 Conclusion

This paper presents MMach, a KHOROS toolbox for image processing by mathematical morphology and several applications of this system in image analysis. MMach increases KHOROS potential by adding a set of high performance tools for multiple purpose use.

The main characteristics of MMach are its hierarchical decomposition structure and the polymorphism associated to its operators. The first property gives high modularity and portability to the software, while the second one permits the representation of abstract concepts (operators defined on general domains) keeping the efficiency of their use.

The implemented elementary operations of mathematical morphology (running on a SPARC station 2) perform a dilation or an erosion on a 256×256 image in about 8 ms in the binary case, and 0.1 s in the gray-scale case.

For each main program of the toolbox there is an online help, which gives the definition of the operator and a set of well-known parameters useful for extracting image information.

Since a high level operator can be built either as a C program or a CANTATA workspace, the toolbox is useful for two main purposes: to solve real image processing problems and didactic applications.

At the moment, we are implementing a new version of MMach that will have an extension of the present set of

operators for 3-D images and 2-D neighborhood graphs, as well as a complementary set of morphological algorithms (granulometries, measures, geodesic distance transform, etc.). An important advancement of the new version of MMach is that it will be an independent library that could have several different interfaces for Khoros, Matlab, Mathematica, etc.

Additionally, this paper has introduced the elementary operators that perform expansion or shrinking of images and that are useful, for example, in the synthesis of artistic images.

Acknowledgment

The authors thank Prof. Arnaldo Mandel and his team for their help with the use of the IME-USP local network. The authors also thank Adriano N. Rodrigues who has ported a large part of the toolbox in its previous version, and to Prof. Lucio Flavio Cavalcanti Pessoa for the filariosis image.

During the period of elaboration of this work, the authors have obtained partial support of CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico under the AnIMoMat Project, contract 680067/94-9). J. Barrera and R. Hirata Jr. have also obtained partial support from Olivetti do Brasil.

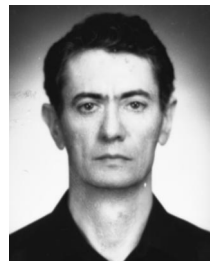
References

1. J. Serra, Ed., *Image Analysis and Mathematical Morphology. II: Theoretical Advances*. Academic Press, London (1988).
2. H. J. A. M. Heijmans, *Morphological Image Operators*, Academic Press, Boston (1994).
3. J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, London (1982).
4. G. J. F. Banon and J. Barrera, "Bases da Morfologia Matemática para Análise de Imagens Binárias," *IX Escola de Computação*, Pernambuco (1994).
5. G. J. F. Banon and J. Barrera, "Minimal representation for translation-invariant set mappings by mathematical morphology," *SIAM J. Appl. Math.* **51**, 1782–1798 (1991).
6. G. J. F. Banon and J. Barrera, "Decomposition of mappings between complete lattices by mathematical morphology: Part I. general lattices," *Signal Processing* **30**, 299–327 (1993).
7. G. J. F. Banon, "Characterization of translation-invariant elementary morphological operators between gray-level images," Research Report INPE-5616-RPQ/671, Instituto Nacional de Pesquisas Espaciais—INPE, CP 515, 12201-970 São José dos Campos, Brazil (November 1995).
8. G. J. F. Banon, "Characterization of translation-invariant elementary morphological operators for gray-level morphology," *Proc. SPIE* **2568** (1995).
9. P. Maragos, "Unified Theory of Translation-Invariant Systems with Applications to Morphological Analysis and Coding of Images," PhD thesis, School of Electrical Engineering, Georgia Institute of Technology, Atlanta (1985).
10. H. J. A. M. Heijmans, "Composing morphological filters," Research report BS-R9504, CWI (1995).
11. T. Y. Kong and A. Rosenfeld, "Digital topology: Introduction and survey," *Computer Vision, Graphics and Image Processing* **48**, 357–393 (1989).
12. J. Serra and L. Vincent, "An overview of morphological filtering," *IEEE Transactions on Circuits, Systems and Signal Processing* **11**, 47–108 (1992).
13. L. Vincent, "Morphological grayscale reconstruction in image analysis: efficient algorithms and applications," *IEEE Transactions on Image Processing* **2**, 176–201 (1993).
14. J. Barrera and G. Pablo Salas, "Set operations on closed intervals and their applications to the automatic programming of morphological machines," *J. Electronic Imaging* **5**, 335–352 (1996).
15. S. Beucher and F. Meyer, "The morphological approach to segmentation: the watershed transformation," *Mathematical Morphology in Image Processing*, in E. R. Dougherty, Ed., Chap. 12, pp. 433–481, Marcel Dekker, New York (1993).
16. B. K. Jang and R. T. Chin, "Analysis of thinning algorithms using mathematical morphology," *IEEE Transactions on Pat. Anal. Mach. Intel.* (1990).
17. J. Rasure, D. Argiro, T. Sauer, and C. Williams, "Visual language and software development environment for image processing," *International Journal of Imaging Systems and Technology* **2**, 183–199 (1990).
18. K. Konstantinides and J. Rasure, "The KHOROS software development environment for image and signal processing," *IEEE Trans. Image Processing* **3**(3), 243–252 (1994).
19. S. Koehnner, Rasure, and Griffey, "Clustering and classification of multi-spectral magnetic resonance images," *Third IEEE Computer-Based Medical Systems Symposium* (1990).
20. R. Sauer and Gage, "Near ground level sensing for spatial analysis of vegetation," *IEEE Conf. Multisource Data Integration in Remote Sensing* (NASA pub. 3099) (1990).
21. S. Kluth, Henning, and Winsand, "Detecting man-made objects in low resolution sar using fractal texture discriminators," *IEEE Proc. IGARS* (1992).
22. Rots and Hereld, "Prototyping astronomical software in khoros," *Astronomical Data Analysis Software and Systems* (1992).
23. P. Soille, "Morphologie Mathématique du relief à la dimensionnalité - Algorithmes et Methods," PhD thesis, Faculté de Sciences Agronomique de l'Université Catholique de Louvain, Louvain-la-Neuve (1992).
24. L. J. Vliet and J. H. Verwer, "A contour processing method for fast binary neighbourhood operations," *Pattern Recognition Letters* **7**(1), 27–36 (January 1988).
25. P. Soille and L. Vincent, "Determining watersheds in digital pictures via flooding simulations," *Proc. SPIE* **1360**, 240–250 (1990).



Junior Barrera received the degree of Doctor in automatic control and systems from the School of Electrical Engineering of the University of Sao Paulo in 1992. Since June 1992, he has been assistant professor at the Department of Computer Science of the Institute of Mathematics and Statistics of the University of Sao Paulo. In the last ten years he has worked in the field of image analysis by mathematical morphology. He has several international publications with theoretical and applied contributions.

Presently, his main research interest is in the development of formal techniques for the automatic programming of morphological machines.



Gerald Jean Francis Banon received the Docteur Ingénieur and Docteur d'état degree in electronic engineering from the Paul Sabatier University, Toulouse, France, in 1971 and 1977. From 1968 to 1970, he was Professeur Ingénieur at the National Institute of Applied Sciences (INSA), Toulouse, France. From 1971 to 1972, he was associated to the postgraduate program coordination (COPPE) of the University of Rio de Janeiro (UFRJ), Brazil.

From 1972 to 1973, he was a research fellow of the Royal Society at Southampton University, England. From 1973 to 1983, he was researcher at the National Center of Scientific Research (CNRS) at the Automatic Control and System Analysis Laboratory (LAAS), Toulouse, France; he spent the year of 1976 at the Electronics Research Laboratory at the College of Engineering of the University of California, Berkeley, and the years of 1981 and 1982 at the University of Pernambuco, Recife, Brazil. From 1983 to 1986, he was an associated researcher at the IBM Scientific Center in Brasília, Brazil. Since 1986, he has been a senior researcher at the National Space Research Institute (INPE), São José dos Campos, Brazil. His main research interests are presently in the area of image processing and mathematical morphology.



Roberto de Alencar Lotufo received the electronic engineering Diploma from Instituto Tecnológico de Aeronáutica, Brazil, in 1978, the MS degree from State University of Campinas, UNICAMP, Brazil, in 1981, and the PhD degree from the University of Bristol, United Kingdom, in 1990, all in electrical engineering. Since 1981, he has been with the Department of Computer Engineering and Industrial Automation at State University of Campinas (UNICAMP),

Brazil. His principal interests are in the areas of multidimensional image processing and analysis, mathematical morphology, and medical imaging.



Roberto Hirata, Jr. is a PhD student in computer science at the Institute of Mathematics and Statistics of the University of São Paulo. Presently, his main research interest is automatic programming of morphological machines under algebraic constraints. He received his MS in computer science in 1997, a BS in physics and a BS in mathematics by the same university in 1990.