MICROMORPH®

APPLICATIONS

TABLE OF CONTENTS

Introduction	4
Basic notions	6
Notations	6
Definitions	7
Some relations	8
Other definitions	9
Exercises	9
Erosions & dilations	12
Erosions, dilations, reminder	12
Gradients	13
Exercises	14
Summary	28
Openings, closings	30
Openings, closings, definition	30
Top-hat	30
Exercises	31
Summary	45
Morphological filters	46
Reminder	46
Contrasts, reminder	46
Exercises	47
Summary	54
Geodesy and connectivity	56
Reminder, binary case	56
Greytone case	56
Exercises	57
Summary	62
Applications of geodesy	64
Exercises	64
Summary	72
Skeletons	74
Exercises	74
Summary	79

Thinnings and thickenings	80
Introduction	80
Binary thinnings, reminder	80
Homotopy	80
Greytone thinnings	81
Exercises	81
Summary	102
Watershed transformation	104
Zone of influence	104
Watershed transformation	104
Exercises	105
Summary	110
Segmentation	112
Exercises	112
Measures	124
Reminder	124
Measures and morphological transformations	124
Suggestions	124
Exercises	125
Summary	133

INTRODUCTION

This practical manual of Mathematical Morphology (abbreviated MM) is essentially made up of exercises designed to introduce you, step by step, to the main notions of MM. The exercises described here are to be solved with MICROMORPH, a teaching software for PC computer which was developed at the Centre de Morphologie Mathématique (Paris School of Mines). This software allows the user to build up progressively a dictionary of morphological transformations, and so doing teaches him to handle transformations and algorithms of increasing complexity.

It is preferable to perform the exercises in order of presentation. Indeed, in most cases they introduce a new transformation you will have to define and add to the dictionary, to be able to solve the following exercises.

Each chapter of the present manual is devoted to one or more fundamental notions of MM. After a quick reminder of the necessary notions and definitions, the exercises are exposed. Because of the structure of MICROMORPH language, many of them are inspired by algorithmics. However, not to delude the user into thinking that MM is only a sequence of transformations, we added, each time it seemed useful, some exercises which do not resort to MICROMORPH, but rather destined to show either important results and theorems, or the use and information one can draw from the performed transformations. In particular, some exercises form a complete application of MM to the solving of an image analysis problem.

Other exercises should be considered as an extension to the learning of theoretical notions, as it is not possible to have an exhaustive view of MM, for lack of time, and also for not confusing the novice morphologist who would not be able to distinguish between things of primary and secondary importance. Thus, such notions as covariance, linear size distribution, random sets, etc., belong to the "cultural background" of the average morphologist, and therefore must be known.

The exercises of this manual use the images supplied with MICROMORPH software. Though it is indispensable to apply the exercises to the suggested image(s), nothing prevents you from testing your algorithms on images of your own .

The transformations which are designed during these exercises are stored in files with a .mic extension. Many transformations which are not presented in the exercises can also be found in these files. Moreover, some files (utility.mic, display.mic and so on) contain many utilities. So, don't forget to have a look to these files whenever a non documented transformation is used in the following exercises.

Chapter 1

BASIC NOTIONS

In this chapter are reviewed the various notations used throughout this book, together with some elementary definitions. To know how to use MICROMORPH software, see the reference manual.

1.1. Notations

Sets represent binary images, functions (from $\mathbb{R}^2$ into $\mathbb{R}$) represent greytone images.

1.1.1. Sets

Sets are generally denoted with capital letters, X, Y, Z. X_i is the i-th connected component of the set X.

1.1.2. Structuring elements

The capital letters B, H, L, M, T, etc... are used for the structuring elements. T_1 , T_2 are the two components of a two-phase structuring element T = (T_1 ,T_{23}).

1.1.3. Functions

Functions are generally denoted with small letters, f, g, h, etc. .

1.1.4. Points, vectors

Small letters represent points or vectors indifferently. The distinction between the two notations is very easily made when the context is known. For instance:

$x \in X$ (in this case, x is a point included in set X).
$x + y = z$ (x, y and z are three vectors with same origin O).

This notation is equivalent to : $Ox + Oy = Oz$.

1.1.5. Set and function transforms

The Greek capital letters Ψ, Φ etc., denote transforms. Φ(X) (Φ(f) respectively) is the result of the transformation Φ applied to the set X (to the function f respectively).

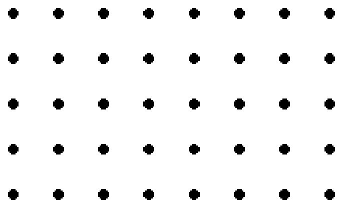
1.1.6. Sub-graph

The sub-graph or umbra of a function f from $\mathbb{R}^n$ into $\mathbb{R}$ is denoted by U(f) and represents the set of the points (x,y) of $\mathbb{R}^n x \mathbb{R}$ such that $y \leq f(x)$.

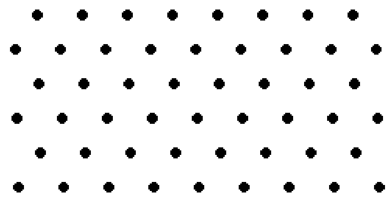
1.2. Definitions

1.2.1. Raster

$\mathbb{Z}$ generally denotes the integer set (i.e. positive, negative numbers or zeros), $\mathbb{Z}^2$ the set of ordered pairs of two integers. $\mathbb{Z}^2$ is called a raster, and its elements are the pixels.



Square grid



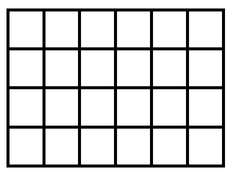
Hexagonal grid

1.2.2. Grid, neighborhood graph

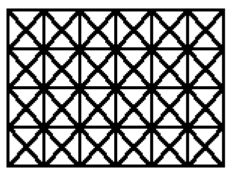
A graph is defined by a set of points which are called the vertices of the graph, and a set of pairs of points taken among the vertices and called the edges of the graph.

A 2-D grid is a graph:

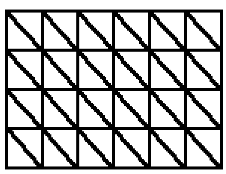
- whose vertices belong to $\mathbb{Z}^2$
- which is invariant under translation in $\mathbb{Z}^2$
- and where the segments joining every edge extremity cannot cross each other.



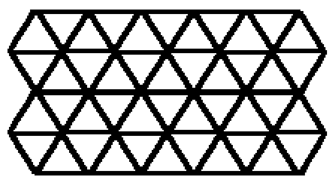
Square graph



Octogonal graph



Hexagonal graph



Hexagonal graph

Apart from a few exceptions, the exercises deal with the hexagonal grid.

1.2.3. Complementation

1.2.3.1. Binary images

X is the complementary set of X. Any point x which is not included in X belongs to X.

1.2.3.2. Greytone images

Image complementation is defined by:

$$f = MAX - f$$

where f is the original image and MAX is the maximum grey level that can be coded with the used system (Up to 32767 with MICROMORPH for Windows).

1.2.4. Union, intersection

1.2.4.1. Binary images

$X \cup Y$ represents the union of the two sets X and Y, that is the set of all the points that belongs to X or to Y. $X \cap Y$ represents the intersection of the two sets X and Y, that is the set of all the points that both belong to X and Y.

1.2.4.2. Greytone images, Sup, inf

Sup(f,g), also denoted $f \vee g$, designates the sup of two functions f and g: in every point x, $(f \vee g)(x)$ is the higher of the two values f(x) and g(x). Inf(f,g), also denoted $f \wedge g$, designates the inf of two functions f and g: in every point x, $(f \wedge g)(x)$ is the smaller of the two values f(x) and g(x).

1.3. Some relations

1.3.1. De Morgan's formulae

These formulae express the duality of union and intersection:

$$(X \cup Y)^c = X^c \cap Y^c \text{ et } (X \cap Y)^c = X^c \cup Y^c$$

Similarly:

$$(f \vee g)^c = f^c \wedge g^c \text{ et } (f \wedge g)^c = f^c \vee g^c$$

1.3.2. Difference, symmetrical difference

X/Y denotes the set difference of X and Y. It is the set of those points belonging to X and not to Y.

$$X/Y = X \cap Y^c$$

X*Y denotes the symmetrical set difference. It is the set of the points that belong to one and only one of the two sets X and Y.

$$X*Y = (X \cap Y^c) \cup (Y \cap X^c) = (X \cup Y)/(X \cap Y)$$

1.3.3 Commutativity, associativity, distributivity

Union, symmetrical difference and intersection are commutative and associative operations. The same properties apply to sup and inf.

commutativity: $X \cup Y = Y \cup X$

$f \vee g = g \vee f$

associativity : $(X \cup Y) \cup Z = X \cup (Y \cup Z) = X \cup Y \cup Z$

$(f \vee g) \vee h = f \vee (g \vee h)$

Union and intersection are mutually distributive:

$$(X \cup Y) \cap Z = (X \cap Z) \cup (Y \cap Z)$$

$$(X \cap Y) \cup Z = (X \cup Z) \cap (Y \cup Z)$$

and so are sup and inf:

$$(f \vee g) \wedge h = (f \wedge h) \vee (g \wedge h)$$
$$(f \wedge g) \vee h = (f \vee h) \wedge (g \vee h)$$

1.4. Other definitions

You will find below a quick reminder of a few definitions which will be useful all along the exercises.

1.4.1. Increasing transformations

A transformation Ψ is said to be increasing if it satisfies:

$$X \subset Y \Rightarrow \Psi(X) \subset \Psi(Y)$$
$$f \leq g \Rightarrow \Psi(f) \subset \Psi(g)$$

1.4.2. Extensivity

Ψ is extensive if:

$$X \subset \Psi(X) \text{ or } f \leq \Psi(f)$$

1.4.3. Idempotence

Ψ is idempotent if:

$$\Psi[\Psi(X)] = \Psi(X) \text{ or } \Psi[\Psi(f)] = \Psi(f)$$

1.4.4. Duality

Φ and $\Psi$are both dual transformations if:

$$\Phi(X) = [\Psi(X^c)]^c \text{ or } \Phi(f) = [\Psi(f)^c]^c$$

EXERCISES

Exercise n° 1

Prove that the function operators inf and sup can be expressed as set operations on sub-graphs. (Prove in particular that the intersection of the sub-graphs f and g is the sub-graph of the inf of f and of g).

Solution

The solution is immediate :

$$\begin{aligned}\forall (x,y), (x,y) \in U(f) \cap U(g) &\Leftrightarrow (x,y) \in U(f) \text{ et } (x,y) \in U(g)\\ &\Leftrightarrow y \leq f(x) \text{ et } y \leq g(x)\\ &\Leftrightarrow y \leq \inf(f(x), g(x))\\ &\Leftrightarrow y \leq \inf(f,g)(x)\\ &\Leftrightarrow (x,y) \in U(\inf(f,g))\end{aligned}$$

Exercise n° 2

1) Prove and verify the above set relations on the binary images provided (and in particular De Morgan's formulae).

2) A transformation Ψ is defined as follows:

$$\Psi(X) = X \cup Y, \text{ Y fixed set}$$

- Is this transformation increasing, extensive, idempotent?
- Does there exist a dual transformation, and if so, indicate it.

Solution

1) De Morgan formulae

Let us prove that :

$$(X \cap Y)^c = X^c \cup Y^c$$

Let x be a point that belongs to the complementary set of $X \cap Y$, x does not belong to $X \cap Y$. But, if x does not belong to both X and Y, it means that it does not belong to one of them at least: $x \notin X$ or $x \notin Y$. Q.E.D.

2) Let $\Psi(X) = X \cup Y$, Y is fixed.

Ψ is increasing:

$$\forall X_1 \subset X_2,\ \Psi(X_1) = X_1 \cup Y \subset X_2 \cup Y = \Psi(X_2)$$

Ψ is extensive (obvious).

Ψ is idempotent:

$$\Psi[\Psi(X)] = (X \cup Y) \cup Y = X \cup Y = \Psi(X)$$

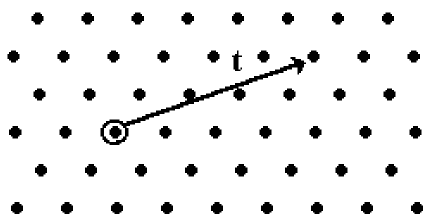
If there exists a dual transformation Φ of Ψ, it satisfies:

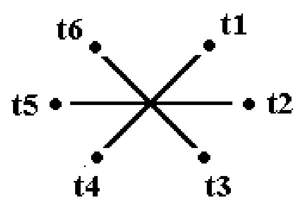
$$[\Psi(X^c)]^c = \Phi(X), \text{ that is:}$$

$$\Phi(X) = (X^c \cup Y)^c = X \cap Y^c$$

Exercise n° 3

On the hexagonal grid each point has six neighbors. Translations are easily defined on this grid.





Each translation is composed of elementary ones, e.g.:

$$t = t_1 \circ t_1 \circ t_2 \circ t_2 \circ t_2$$

The set of translations equipped with the composition law q constitutes the group of displacements.

Do the shift operators of MICROMORPH satisfy this group structure? Verify and explain your answer. (This phenomenon is the first occurrence of the border effects resulting from the fact we work on a finite field of analysis).

Solution

(obvious).

As a reminder, border effects are due to the fact that MICROMORPH assumes that the points liable to fall outside the field are forced to zero. This convention will have important consequences for the definition of geodesic transforms, but also for classical Euclidean transforms. This will clearly appear in the following exercises.

Chapter 2

EROSIONS & DILATIONS

2.1. Erosions, dilations, reminder

2.1.1. Binary case

The dilation of a set X by a set B of center O called a structuring element is a set Y thus defined:

$$Y = X \oplus \check{B} = \{x \ : B_x \cap X = \emptyset \}$$

where $B_x = \left\{ x + \overrightarrow{Ob}, b \in B \right\}$.

$(X \oplus \check{B})$ can also be written:

$$X \oplus \check{B} = \bigcup_{b \in \check{B}} X_{\overrightarrow{Ob}}$$

$\check{B}$ is the transposed set of B (i. e. the symmetrical set of B with respect to O). The dilated set Y is then the union of the translates of X.

Similarly, the eroded set is defined by:

$$Z = X \ominus \check{B} = \{x \ : B_x \subset X\}$$

which is also written:

$$X \ominus \check{B} = \bigcap_{b \in \check{B}} X_{\overrightarrow{Ob}}$$

These two transformations have the following properties:

$$(X \cup Y) \oplus \check{B} = (X \oplus \check{B}) \cup (Y \oplus \check{B}) \qquad (1)$$

$$(X \cap Y) \ominus \check{B} = (X \ominus \check{B}) \cap (Y \ominus \check{B})$$

$$(X \oplus \check{B}_1) \oplus \check{B}_2 = X \oplus (\check{B}_1 \oplus \check{B}_2)$$

$$(X \ominus \check{B}_1) \ominus \check{B}_2 = X \ominus (\check{B}_1 \oplus \check{B}_2)$$

which hold, whichever are the centers of B, B_1 et B_2 .

2.1.2. Greytone case

The sub-graph U(f) of a function f can be eroded and dilated by a three-dimensional structuring element B and it can be shown, under certain conditions, that the dilation (resp. the erosion) of a sub-graph is still the sub-graph of a function, called the dilation (resp. the erosion) of f by B and denoted $f \oplus \check{B}$ (resp. $f \ominus \check{B}$).

If the structuring elements are planar, the dilation of a function f can be written:

$$(f \oplus \check{B})(x) = \sup_{b \in \check{B}} \left(f\left(x + \overrightarrow{Ob} \right) \right)$$

or else:

$$f \oplus \check{B} = \sup_{b \in \check{B}} f_{\overrightarrow{Ob}}$$

where $f_{\overrightarrow{Ob}}$ is the translated function of f of vector $\overrightarrow{Ob}$.

Similarly, the erosion is defined by:

$$(f \ominus \check{B})(x) = \inf_{b \in \check{B}} \left(f\left(x + \overrightarrow{Ob} \right) \right)$$

or else:

$$f \ominus \check{B} = \inf_{b \in \check{B}} f_{\overrightarrow{Ob}}$$

The relations (1) are immediately transposable to functions.

$$(f \vee g) \oplus B = (f \oplus B) \vee (g \oplus B) \qquad (1')$$
$$(f \wedge g) \ominus B = (f \ominus B) \wedge (g \ominus B)$$
$$(f \oplus B_1) \oplus B_2 = f \oplus (B_1 \oplus B_2)$$
$$(f \ominus B_1) \ominus B_2 = f \ominus (B_1 \oplus B_2)$$

2.2. Gradients

2.2.1. Classical gradient

The gradient of a function f defined on $\mathbb{R}$ is defined as the vector:

$$\nabla f = \begin{pmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{pmatrix}$$

In the digital case, the first-order differences may be used to express the partial derivatives:

$$(\nabla_x f)(x, y) \approx f(x, y) - f(x-1, y)$$
$$(\nabla_y f)(x, y) \approx f(x, y) - f(x, y-1)$$

Digital convolutions of f are recognized by the kernels [-1 1] and $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$.

We could also use the following differences as the expression of the partial derivatives:

$$(\nabla_{2x} f)(x, y) \approx f(x+1, y) - f(x-1, y)$$
$$(\nabla_{2y} f)(x, y) \approx f(x, y+1) - f(x, y-1)$$

They have the advantage of being centered in (x,y). These derivatives are performed with the digital convolutions of f by the kernels [-1 0 1] and $\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$.

In practice, other differences are also used, which may be expressed in terms of digital convolutions too. An abundant literature illustrates this subject. For example, the Sobel operator is given for the following convolutions:

$$\tfrac{1}{4}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } \tfrac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & -1 \end{bmatrix}$$

2.2.2. Morphological gradient

The Beucher gradient of a function f defined on $\mathbb{R}^2$ or on a sub-set is given by:

$$g(f) = \lim_{\lambda \to 0} \frac{(f \oplus \lambda B) - (f \ominus \lambda B)}{2\lambda}$$

where λB denotes a ball of radius λ.

On a hexagonal grid, we get:

$$g(f) = \frac{(f \oplus B) - (f \ominus B)}{2}$$

where H is a hexagon of size 1. This gradient is equal to the gradient module of a function f continuously derivable.

EXERCISES

Exercise n° 1

1) Prove relations (1) and (1').

2) Prove or invalidate the following statements:

- Erosion and dilation are increasing,
- extensive, anti-extensive,
- idempotent,
- dual.

Solution

1) Let us prove that:

$$(X \cup Y) \oplus \check{B} = (X \oplus \check{B}) \cup (Y \oplus \check{B})$$

$$(X \cup Y) \oplus \check{B} = \bigcup_{b \in \check{B}} (X \cup Y)_b = \bigcup_{b \in \check{B}} (X_b \cup Y_b) = \left(\bigcup_{b \in \check{B}} X_b \right) \cup \left(\bigcup_{b \in \check{B}} Y_b \right)$$

that is $(X \oplus \check{B}) \cup (Y \oplus \check{B})$.

The second relation can be proved in the same way. Let us prove the third relation :

$$(X \oplus \check{B}_1) \oplus \check{B}_2 = \bigcup_{b_2 \in \check{B}_2} (X \oplus \check{B}_1)_{b_2} = \bigcup_{b_2 \in \check{B}_2} \left(\bigcup_{b_1 \in \check{B}_1} X_{b_1} \right)_{b_2} = \bigcup_{b_1 \in \check{B}_1, b_2 \in \check{B}_2} (X_{b_1 + b_2})$$

$$= \bigcup_{b \in \check{B}_1 \oplus \check{B}_2} (X_b) = X \oplus (\check{B}_1 \oplus \check{B}_2)$$

2) Let us verify the statements:

- Erosion and dilation are increasing transformations: true (since set union and set intersection are increasing).
- Extensivity or anti-extensivity?

Consider the case of erosion.

$$X \ominus \check{B} = \bigcap_{b \in \check{B}} X_b$$

Since the eroded set is the intersection of the translates of X, it does not contain X, unless B is reduced to the point of origin.

Conversely, stating that $X \ominus \check{B} \subset X$, requires to be able to write:

$$X \ominus \check{B} = X \cap \left(\bigcap_{b \in \check{B} - \{\emptyset\}} X_b \right)$$

which is right only when the structuring element B contains its origin. As a general rule, erosion is then neither extensive nor anti-extensive.

It is the same for dilation.

- Erosion and dilation are idempotent : false (obvious).
- Erosion and dilation are dual transforms:

$$(X \oplus \check{B})^c = \left(\bigcup_{b \in \check{B}} X_b \right)^c = \bigcap_{b \in \check{B}} (X_b)^c = \bigcap_{b \in \check{B}} (X)_b^c = (X^c \ominus \check{B}) \text{ Q.E.D.}$$

Exercise n° 2

1) Use the primitives of the MICROMORPH language to define the following transformations:

- erosion and dilation by a segment of size n (consisting of n+1 consecutive points) in both binary and greytone cases.

$$X \ominus \check{L}_n,\ X \oplus \check{L}_n,\ f \ominus \check{L}_n,\ f \oplus \check{L}_n$$

- erosion and dilation by a pair of points at a distance n:

$$X \ominus \check{K}_n,\ X \oplus \check{K}_n,\ f \ominus \check{K}_n,\ f \oplus \check{K}_n$$

(the origin of the two structuring elements is arbitrarily chosen at one extremity)

2) Prove that:

$$X \ominus \check{L}_n \subset X \ominus \check{K}_n$$

[procedures **direro** ; **dirdil** ; **dblero** ; **dbldil**]

Solution

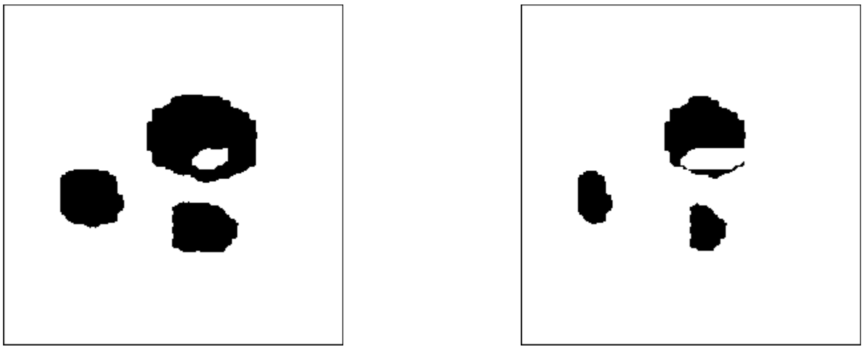
1) Let us define the following transforms:

- Linear binary and greytone erosions:

$X \ominus \check{L}_n$ et $f \ominus \check{L}_n$

$\underbrace{(0.........)}_{\text{n points}}$

```
deproc direro direro dir s d sz
syntax "direro imin imout dir size -> directional line erosion of size sz "
  imcopy s d
  iminfngb d d dir sz
end
```



(a) (b)

Linear binary erosion

(a) original, (b) horizontal erosion of size 23

- Linear binary and greytone dilations:

$X \oplus \check{L}_n$ et $f \oplus \check{L}_n$

```
deproc dirdil dirdil dir s d sz
syntax "dirdil dir imin imout size --> directional line dilation"
    imcopy s d
    imsupngb d d dir sz
end
```

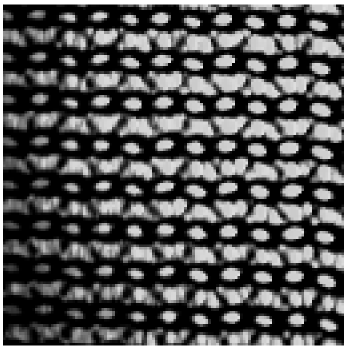
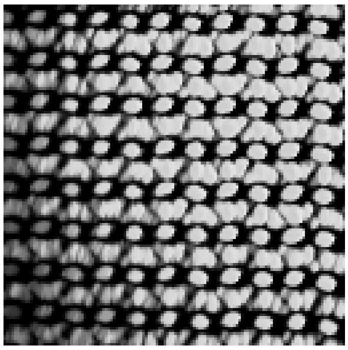
- Erosion by a pair of points:

$X \ominus \check{K}_n$ et $f \ominus \check{K}_n$

$\underbrace{(0\qquad\qquad .)}_{\text{distance n}}$

```
deproc dblero dblero dir s d sz
```

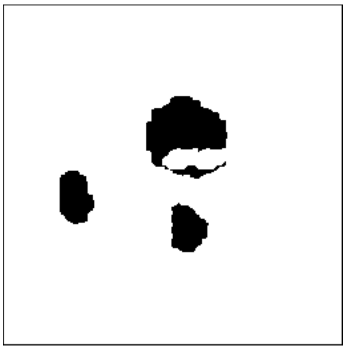
```
syntax "greydblero dir greyin greyout size"
  int w ;
  w := imalloc imdepth s
  imcopy s w
  imcopyngb w w dir sz impixmin s
  iminf s w d
  imfree w
end
```

(a) (b)

Linear greytone dilation
(a) original, (b) linear dilation of size 5

This transformation uses a working memory. The transformation is called covariance in M.M. (see further).



Covariance of size 23

- Dilation by a pair of points:
$X \oplus \check{K}_n$, $f \oplus \check{K}_n$

```
deproc dbldil dbldil dir s d sz
syntax "greydbldil dir greyin greyout size"
  int w ;
  w := imalloc imdepth d
  imcopy s w
  imcopyngb w w dir sz impixmin s
  imsup s w d
  imfree w
```

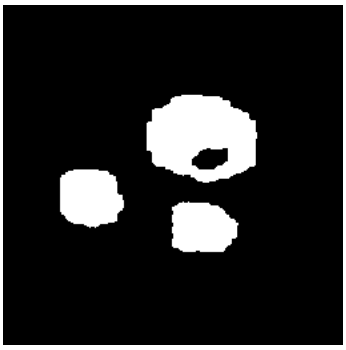
end

2) Prove that $X \ominus \check{L}_n \subset X \ominus \check{K}_n$. Let x be a point of $X \ominus \check{L}_n$. The structuring element L_n translated in x is then included in X, and so are its two extremities consequently. Then x belongs to the eroded class $X \ominus \check{K}_n$ (Q.E.D.)

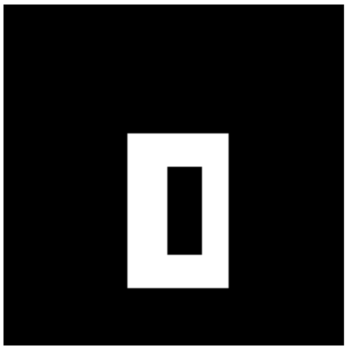
Exercise n° 3

1) Prove that an elementary hexagon may be generated by three successive dilations of a point by three judiciously chosen segments. Then, deduce an algorithm that allows to obtain the erosion and the dilation by an hexagon. Program these transformations (both for the hexagonal and square grids), and verify your algorithms on the provided binary images CAT, OBJECTS, SHAPE and greytone images KNITTING, SALT and CIRCUIT.

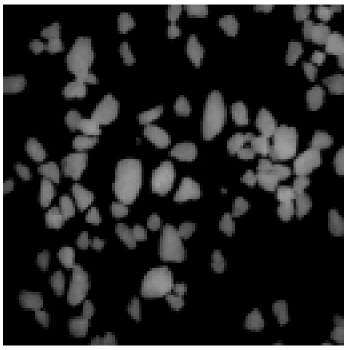
CAT



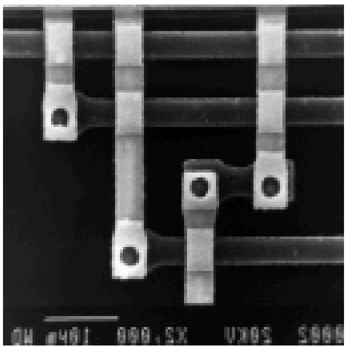
OBJECTS



SHAPE



SALT



CIRCUIT

2) Verify the duality of these transformations. Is the result satisfactory?
[Procedures **ero** ; **dil**]

Solution

1) The figure below shows how to obtain a hexagon from three segments. We have:

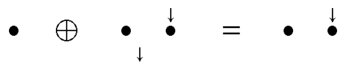
$$H = L1 \oplus L2 \oplus L3$$

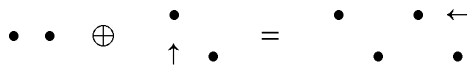
Then, we can write:

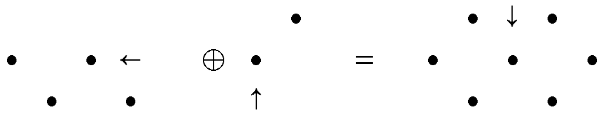
$$X \oplus H = ((X \oplus L1) \oplus L2) \oplus L3$$

and also:

$$X \ominus H = X \ominus (L1 \oplus L2 \oplus L3) = ((X \ominus L1) \ominus L2) \ominus L3$$


(segment L1)


(segment L2)


(segment L3)

The same procedure can be used, with a fourth direction, for defining a square erosion. This erosion can be defined this way:

```
deproc ero ero s d sz
syntax "ero imin imout size"
   imcopy s d
   if (grid = 1) then
     for 1 to sz do
       iminfngb d d 1 1
       iminfngb d d 3 1
       iminfngb d d 5 1
     end
  else
     for 1 to sz do
       iminfngb d d 1 1
       iminfngb d d 3 1
       iminfngb d d 5 1
       iminfngb d d 7 1
     end
 end
```

(a) initial picture
(b) linear dilation at 60° . The point exits the field and is lost.

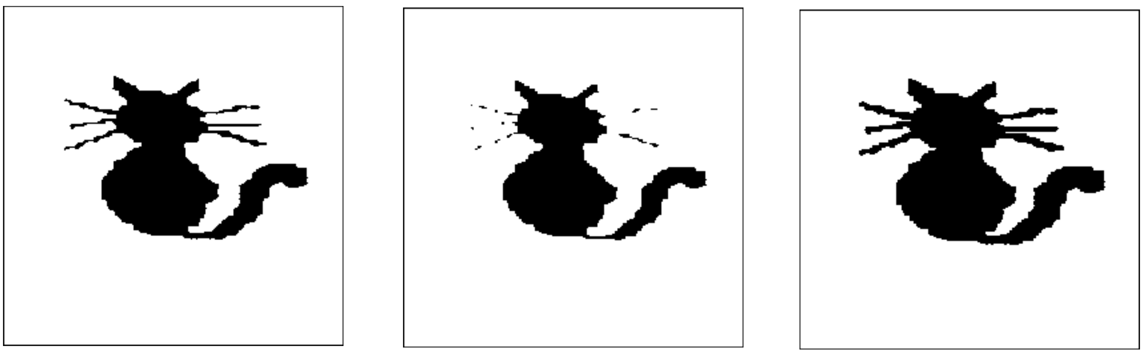
We could use the same formula to define hexagonal dilation. However, it would not take into account border effects. Indeed, a point which is on the field border will not always be correctly dilated, because linear dilation may propagate a white point outside the field. This white point is then irremediably lost.

Note that this problem does not occur with the square grid.

The solution consists in performing translations in the six directions of the hexagonal grid.

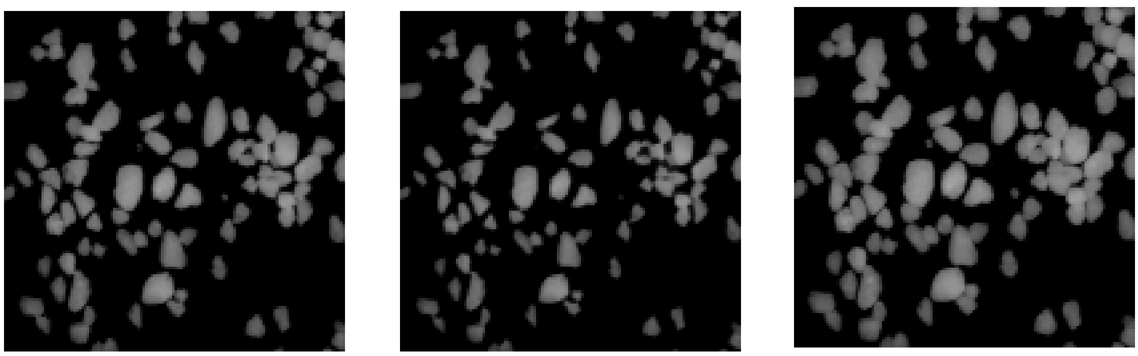
```
deproc dil dil s d sz
syntax "dil in out size"
   int w i ;
  imcopy s d
```

```
if (grid = 1) then
  w := imalloc imdepth s
  for 1 to sz do
    i := 0
   imcopy d w
    for 1 to 6 do
      imsupngb d w ++ i 1
    end
   imcopy w d
  end
  imfree w
 else
  for 1 to sz do
    imsupngb d d 1 1
    imsupngb d d 3 1
    imsupngb d d 5 1
    imsupngb d d 7 1
  end
 end
end
```



a) (b) (c)

Hexagonal binary erosion and dilation, (a) initial picture, (b) erosion, (c) dilation.



(a) (b) (c)

Hexagonal greytone erosion and dilation, (a) initial picture, (b) erosion, (c) dilation

2) Verification of duality (see important note below)

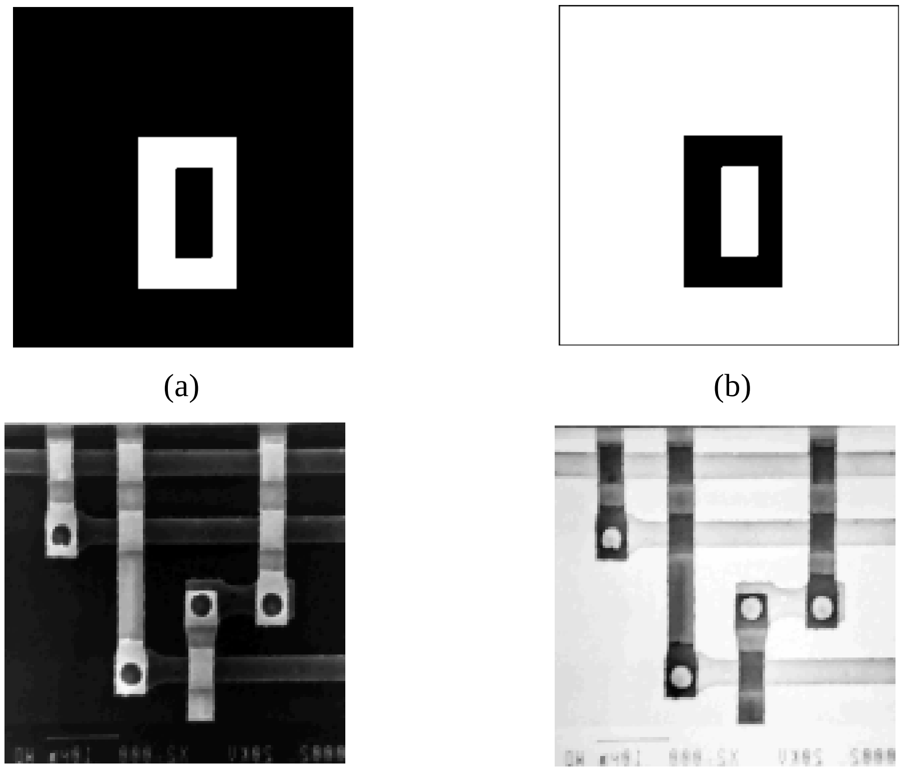
Use any image (take a binary one for an easier display of the phenomenon) stored in memory 1, and apply:

dil b1 b2 1
imdisplay b2 "dilaté"

Next:

iminv b1 b3
ero b3 b3 1
iminv b3 b3
imdisplay b3 "dualité"

Note the appearance of a frame on the second image, due to border effects (the inversion only works within the image field).



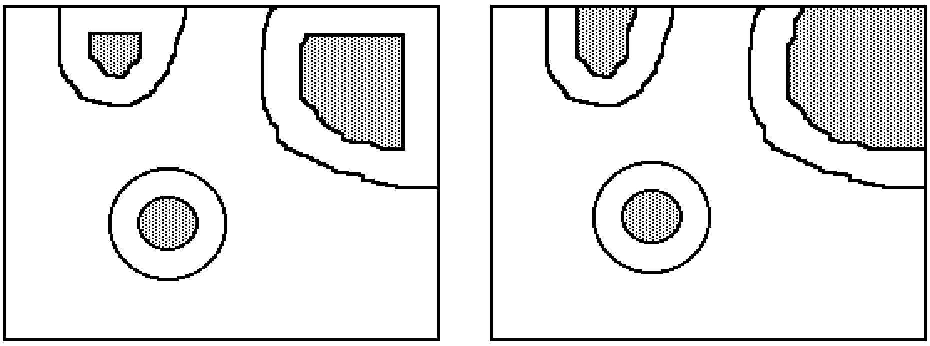
(a) (b) (c) (d)

Verification of duality
(a) erosion of a binary image, (b) complementary dilation
(c) erosion of a greytone image, (d) complementary dilation

WARNING! Elementary transformations and border effects

It is important, for obtaining the results given in the previous exercises, to set to 0 the **edge** variable. If it is not the case, enter the command **imsetedge 0** before trying the transformations. You will see, in this case, that binary or greytone erosions produce a dark border. In this mode, the outside of the image is considered as being empty. The objects under study are entirely included in the field of analysis. However, this mode is penalizing, especially for greytone images because it deeply reduces the image field when the size of the transformation increases. Therefore, there exists another mode called "standard mode" (It is, in fact a geodesic mode, see chapter on the geodesic transforms) where the structuring element B when crossing the boundary of the image field D does not take into account its points which are outside D. This is equivalent to perform the transformation with the structuring element$B \cap D$.

In case of dilation, this mode and the previous one are the same. But it is not true for the erosion as illustrated below:



Difference between erosion in mode 0 (left) and in "standard" mode (right)

To realize this transformation, one could think that setting **edge** to 1 by means of the statement **imsetedge 1** would be sufficient. Unfortunately, in the hexagonal case the erosion presents the same defects as the dilation when this latter transformation is made using only three directions. Therefore, it is necessary, in order to obtain unbiased transformations at the edges whatever the selected mode, to perform the hexagonal erosion and dilation by means of translations in the six main directions of the grid. The corresponding procedure is the following:

```
deproc ero ero s d sz
syntax "ero in out size"
  int w i ;
  imcopy s d
  if (grid = 1) then
    w := imalloc imdepth s
    for 1 to sz do
      i := 0
      imcopy d w
      for 1 to 6 do
        iminfngb d w ++ i 1
      end
      imcopy w d
    end
    imfree w
  else
    for 1 to sz do
      iminfngb d d 1 1
      iminfngb d d 3 1
      iminfngb d d 5 1
      iminfngb d d 7 1
    end
  end
```

```
end
```

This procedure is correct for the hexagonal grid and when **edge** is set to 1, it fulfils the duality rule for the complementation. Its sole drawback comes from the fact that it is slower and it requires more memory resources.

The correct morphological transformations **ero** and **dil** have been implemented in MICROMORPH. However, fast hexagonal erosion and dilation (using only three directions) can be found in the file **ch2misc.mic**. They produce biased results on the edges but nevertheless, they can be used when the speed of the treatment is more important than the accuracy of the result. These two procedures are named **erode** et **dilate**. To use them instead of **ero** and **dil**, just rename them and compile the file. These procedures are given below (remind that they provide no speed enhancement with the square grid):

```
deproc erode erode s d sz
    syntax "erode imin imout size"
      imcopy s d
      if (grid = 1) then
        for 1 to sz do
          iminfngb d d 1 1
          iminfngb d d 3 1
          iminfngb d d 5 1
        end
     else
      for 1 to sz do
        iminfngb d d 1 1
        iminfngb d d 3 1
        iminfngb d d 5 1
        iminfngb d d 7 1
      end
    end
```

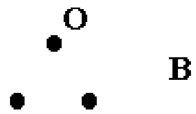
(it's in fact the erosion initially defined at the beginning of the exercise).

```
deproc dilate dilate s d sz
    syntax "dilate imin imout size"
      imcopy s d
      if (grid = 1) then
        for 1 to sz do
          imsupngb d d 1 1
          imsupngb d d 3 1
          imsupngb d d 5 1
        end
     else
      for 1 to sz do
        imsupngb d d 1 1
        imsupngb d d 3 1
        imsupngb d d 5 1
        imsupngb d d 7 1
      end
```

end

Exercise n° 4

1) Program the erosion by an elementary triangle, pointing upwards (take the point for origin).



Program also the erosion by a 2x2 square.

2) What happens when the transformation is iterated? (Perform $(X \ominus \check{B}) \ominus \check{B}$). Represent the structuring element B' such that:

$$(X \ominus \check{B}) \ominus \check{B} = X \ominus \check{B}'$$

A simple way to know how it looks is to dilate a point by B, twice. Indeed:

$$[\{.\} \oplus \check{B}] \oplus \check{B} = \check{B} \oplus \check{B} = \overline{B \oplus B}^{\vee}$$

$$(X \ominus \check{B}) \ominus \check{B} = X \ominus (\check{B} \oplus \check{B}) \Rightarrow B' = B \oplus B$$

3) Program the dilation by a triangle pointing downwards.

4) Perform the dilations by the following structuring elements:

If H is the elementary hexagon, find the structuring element which is equivalent to:

$$B_1 \oplus H \oplus B_2$$

[procedures **miniero** ; **minidil** ; **binb1dil** ; **binb2dil**]

Solution

1) Erosion by a triangle (point upwards)

This erosion is obtained by intersecting the two erosions by segments oriented in the directions 3 and 4.

The erosion by an elementary square is similar (directions 3 and 5).

```
deproc miniero miniero s d
syntax "miniero imin imout"
  int w;
  w := imalloc imdepth s
  imcopy s w
  if (grid = 1) then
    iminfngb s w 3 1
    iminfngb s w 4 1
  else
    iminfngb s w 3 1
    iminfngb w w 5 1
```

```
  end
  imcopy w d
  imfree w
end
```

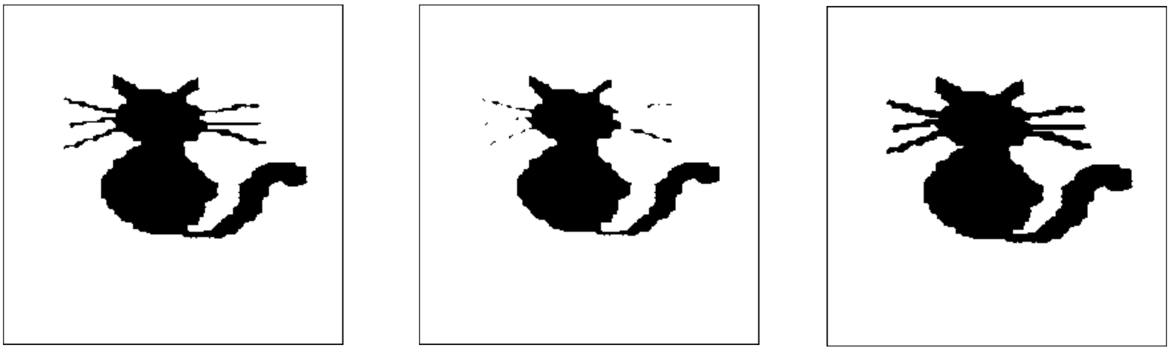
2) Repeating two triangular transformations results in a triangular transformation of size 2.

3) Dilation by a triangle (point downwards)

This dilation is obtained by the union of two linear dilations by segments oriented in the directions 1 and 6.

The following procedure is the dilation by the transposed elementary square.

```
deproc minidil minidil s d
syntax "minidil imyin imout"
   int w ;
   w := imalloc imdepth s
    imcopy s w
    if (grid = 1) then
      imsupngb s w 1 1
      imsupngb s w 6 1
    else
      imsupngb s w 7 1
      imsupngb w w 1 1
    end
    imcopy w d
    imfree w
  end
```



(a) (b) (c)

Triangular erosion, dilation

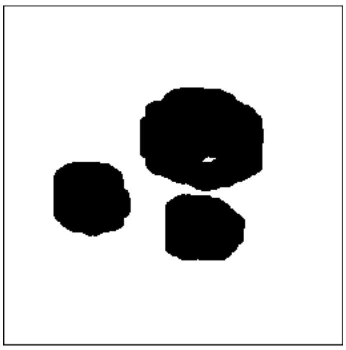
(a) initial binary image, (b) triangular erosion, (c) triangular dilation

4) Dilations using structuring elements B1 and B2

The binary and greytone procedures are the following :

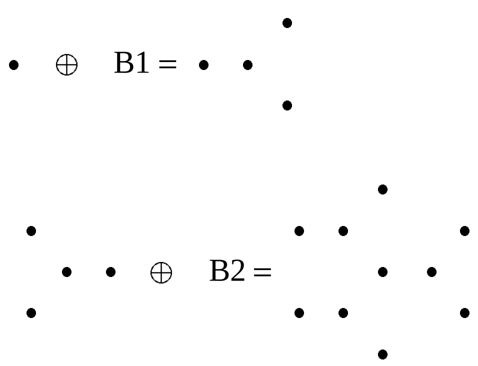
```
deproc b1dil b1dil s d
syntax "b1dil imin imout"
  int w;
  if (grid <> 1) then makeerror 10001 end
```

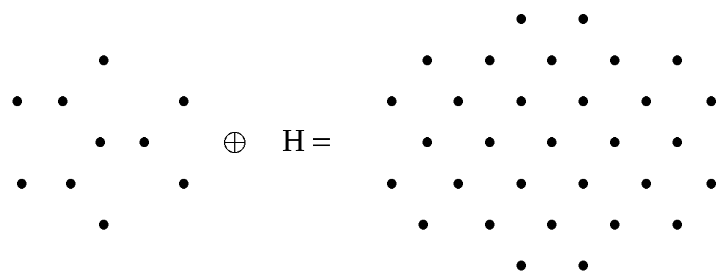
```
  w := imalloc imdepth s
   imcopy s w
   imsupngb s w 1 1
   imsupngb s w 3 1
   imsupngb s w 5 1
   imcopy w d
   imfree w
end

deproc b2dil b2dil s d
  syntax "b2dil imin imout"
   int  w ;
   if (grid <> 1) then makeerror 10001 end
   w := imalloc imdepth s
   imcopy s w
   imsupngb s w 4 1
   imsupngb s w 4 1
   imsupngb s w 4 1
   imcopy w d
   imfree w
 end
```



Dodecagonal dilation

Three successive dilations by B1, B2 and H are equivalent to a dodecagonal dilation.





Exercise n° 5 : Distance function

Let d be a distance defined on the points of the Euclidean grid (length of the shortest path drawn on the grid between two points). At any point x of X, the value of the distance function is $\mathrm{dist}(x) = d(x, X^c) = \min_{y \in X^c} d(x, y)$.

Prove that if d is the distance defined on the hexagonal grid as the length of the shortest path drawn on the grid between two points, the distance function can be obtained by means of the successive erosions of X by a hexagon.
[procedure **distance**]

Solution

For this distance, the set of the points located at a distance smaller than or equal to n from a point x is the hexagon of size n centered in x.

Let x be a point of X located at a distance n from X^c (the distance function in x is n). The hexagon of size n centered in x contains a point of X^c (by hypothesis) and the hexagon of size n-1 centered in x does not contain any point of X^c (otherwise the distance would be smaller than n). Then x belongs to the set X eroded by a hexagon of size n-1and does not belong to the set X eroded by a hexagon of size n.

From which is derived the algorithm for determining the distance function:

```
deproc distance distance s d
syntax "distance s d"
   int w ;
   w := imalloc 1
   imcopy s w
   imset impixmin d d
   while ( imvolume w ) do
      imadd w d d
      ero w w 1
   end
   imfree w
end
```

Exercise n° 6

On the ROAD image, solve the following problems:

1) Program and apply the Sobel gradient.

2) Program and apply the Beucher gradient of size λ of a function f:

$$g(f) = (f \oplus \lambda B) - (f \ominus \lambda B)$$

where λB denotes a ball of size λ (take here B = H).
[procedures **gradient** ; **sobel**]

ROAD

Solution

1) The convolution operator generates a signed image (a pseudo-signed image actually, the positive values being coded from 0 to 127 and the negative values from 128 to 255). Let us define the function allowing to find the absolute value of the image.

```
deproc greyabs greyabs s d
syntax "greyabs greyin greyout"
  int w;
  if (imdepth s = 8) then
    w := imalloc imdepth s
    imcopy s w
    imcsub s 127 w
    imsub s w d
  else
    if (imdepth s = 16) then
      imabs16 s d
    else
      makeerror 1404
    end
  end
  imfree w
end
```

The result is an image whose values are comprised between 0 and 128 (8 bits image).

The Sobel gradient operator is given by :

```
deproc sobel sobel s d
syntax "sobel greyin greyout"
  int w;
  w := imalloc 8
  imconvolve s w  "sobelx"
  greyabs w w
  imconvolve s d "sobely"
  greyabs d d
  imsup w d d
  imfree w
```

```
end
```

2) The "thick" morphological gradient is programmed as follows :

```
deproc gradient gradient s d sz
syntax "gradient greyin greyout size : gradient"
  int w ;
  w := imalloc imdepth s
  dil s w sz
  ero  s d sz
  imsub w d d
  imfree w
end
```

SUMMARY

By the end of this session of exercises, your MICROMORPH dictionary should contain all the transformations listed below. The syntax below is of course not compulsory. However, you are well advised to use it for the readability of the present manual and that of the solutions. At the end of each chapter of exercises, you will find a similar list. Read it before the programming of the transformations so as to spare the tedious work of matching your definitions with those of the list.

direro dir s d sz
erosion by a segment of size sz in a direction dir of the image s into the image d.
dirdil dir s d sz
dilation by a segment of size sz in a direction dir of the image s into the image d.
dblero dir s d sz
erosion by a pair of points at a distance sz in a direction dir of the image s into the image d.
dbldil dir s d sz
dilation by a pair of points at a distance sz in a direction dir of the image s into the image d.
ero s d sz
hexagonal or square erosion of size sz of the image s into the image d.
dil s d sz
hexagonal or square dilation of size sz of the image s into the image d.
miniero s d
erosion by a triangle (pointing upwards) or a 2x2 square of the image s into the image d.
minidil s d
dilation by a triangle (pointing downwards) or a 2x2 square of the image s into the image d.
b1dil s d
dilation by the structuring element B_1 of the image s into the image d.
b2dil s d
dilation by the structuring element B_2 of the image s into the image d.
distance s d
hexagonal distance function by successive erosions of the binary image s into the greytone image d.
sobel s d
Sobel gradient of the greytone image s into the greytone image d.

gradient s d sz
morphological gradient by a hexagon of size sz of the greytone image s into the greytone image d.

Chapter 3

OPENINGS, CLOSINGS

3.1. Openings, closings, definitions

3.1.1. Binary case

Erosion and dilation lead to the definition of two new transformations, the opening γ and the closing φ :

$$\gamma(X) = (X \ominus \check{B}) \oplus B$$
$$\varphi(X) = (X \oplus \check{B}) \ominus B$$

3.1.2. Greytone case

The opening and the closing are defined similarly:

$$\gamma(f) = (f \ominus \check{B}) \oplus B$$
$$\varphi(f) = (f \oplus \check{B}) \ominus B$$

3.1.3. Properties, size distribution

The opening is an increasing and anti-extensive operation. The closing is increasing and extensive. The two transformations are idempotent.

If λB denotes the homothetic set of a convex set B, the opening is a size distribution. In particular:

$$\text{si } \lambda > \mu,\ [(X)_{\lambda B}]_{\mu B} = \left[(X)_{\mu B}\right]_{\lambda B} = (X)_{\lambda B}$$
$$[(f)_{\lambda B}]_{\mu B} = \left[(f)_{\mu B}\right]_{\lambda B} = (f)_{\lambda B}$$

The opening and the closing are used for size distribution analysis on the one hand, and for filtering on the other hand. These two kinds of use are illustrated in the exercises.

3.2. Top-hat

The top-hat is a transformation that only applies to greytone images. The top-hat WTH of a function f is defined by:

$$WTH(f) = f - \gamma(f) \quad \text{(white Tophat)}$$

Likewise, the conjugated top-hat BTH of a function f is defined by:

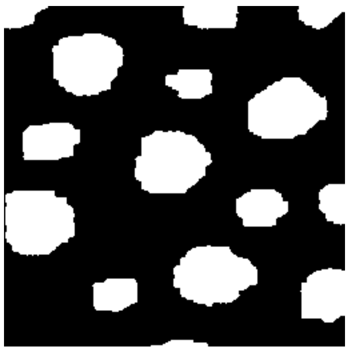
$$BTH(f) = \varphi(f) - f \quad \text{(black Tophat)}$$

EXERCISES

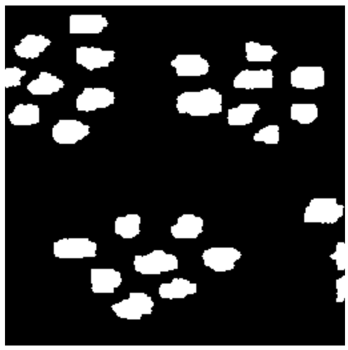
Exercise n° 1

1) In order to program the following transformations, use the ones (erosions, dilations) already generated in the dictionary:
- opening and closing by a hexagon of size n,
- opening and closing by a segment of size n and
- opening and closing by a pair of points at a distance n.

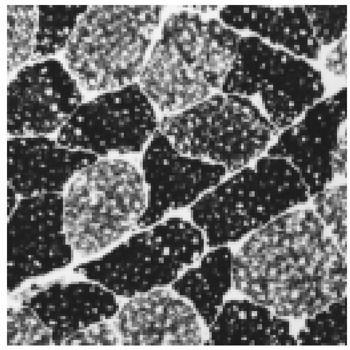
2) Verify the properties stated above. Prove in particular the duality of opening and closing. Verify also that the opening by a pair of points is not a size distribution (openings of small size suppress more points than openings of greater size). Use in particular the images PARTIC2 and KNITTING.
[procedures **open** ; **close** ; **diropen** ; **dirclose**]



PARTIC1



PARTIC2

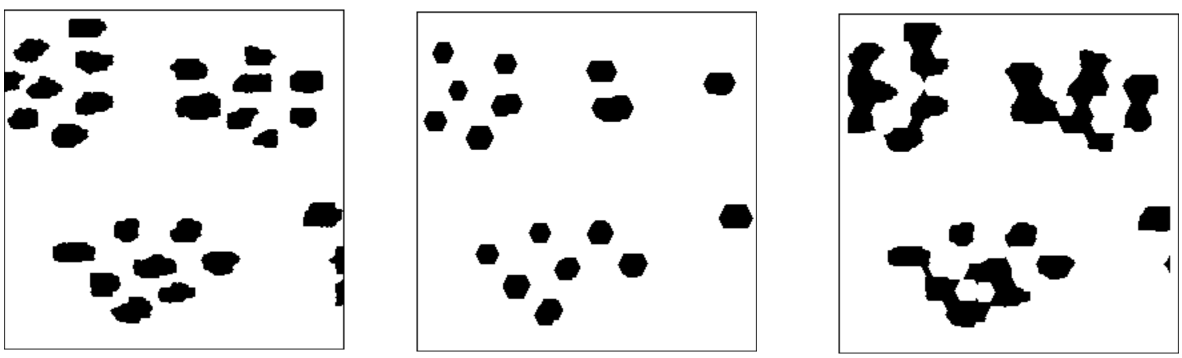


MUSCLE

Solution

1) Definition of the transformations:
- <u>Hexagonal opening and closing</u>:

```
deproc open open s d sz
syntax "open imin imout size"
   ero s d sz
   dil d d sz
  end
```



(a) (b) (c)

binary opening and closing
(a) initial image, (b) (c) hexagonal transformation

```
deproc close close s d sz
syntax "close imin imout size"
  dil s d sz
  ero d d sz
end
```

- Linear opening and closing :

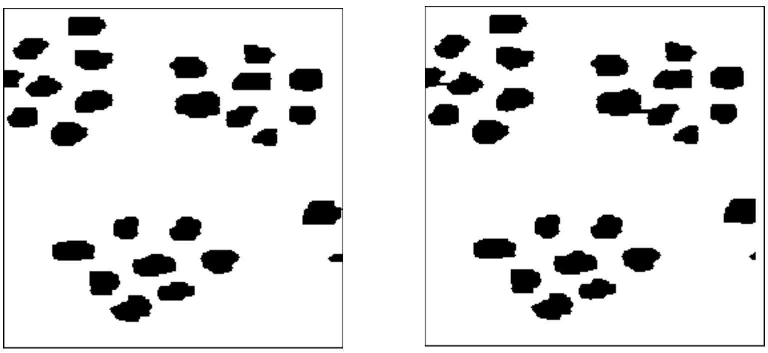
```
deproc diropen diropen dir s d sz
syntax "diropen dir imin imout size"
   direro dir s d sz
   dirdil dirtranspose dir d d sz
end
```

```
deproc dirclose dirclose dir s d sz
syntax "dirclose dir imin imout size"
  dirdil dir s d sz
  direro dirtranspose dir d d sz
end
```

Note the use of the word **dirtranspose** (found in the standard library) which allows to compute the transposed direction of the current structuring element when the latter is not symmetrical.

- Opening and closing by a pair of points:

```
deproc dblopen dblopen dir s d sz
syntax "dblopen dir binin binout size"
 dblero dir s d sz
 dbldil dirtranspose d d sz
end
```



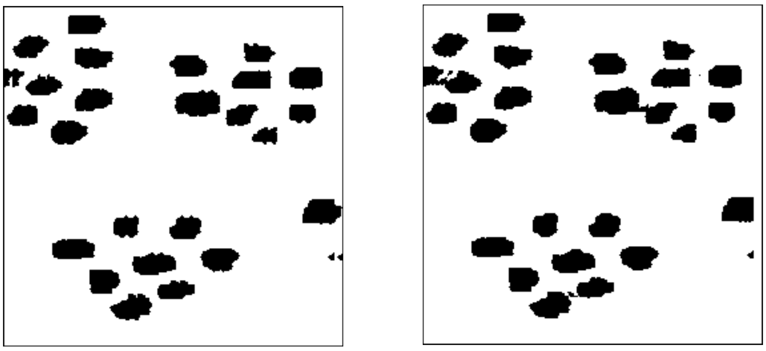
(a) (b)
Binary opening and closing (continued)
(a) linear opening, (b) linear closing

```
deproc dblclose dblclose dir s d sz
syntax "dblclose dir binin binout size"
 dbldil dir s d sz
```

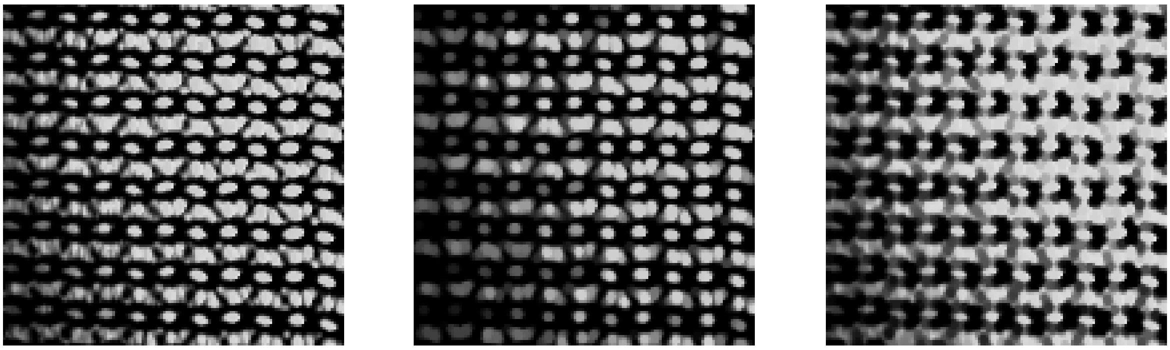
dblero dirtranspose d d sz
end

2) Duality of opening and closing
It is possible to write:

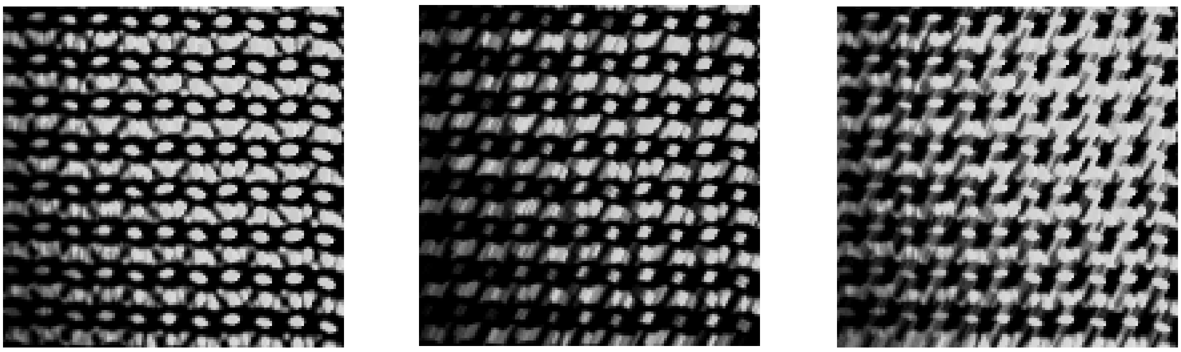
$$(X)_B = (X \ominus \check{B}) \oplus B = (X^c \oplus \check{B})^c \oplus B$$
$$= ((X^c \oplus \check{B}) \ominus B)^c = \left[(X^c)^B\right]^c, \text{ Q.E.D.}$$



(a) (b)
Binary opening and closing (end)
Opening (a), closing (b) by a pair of points



(a) (b) (c)
Greytone opening and closing
(a) initial image, hexagonal opening (b), closing (c)



(a) (b) (c)
Greytone opening and closing (continued)
(a) initial image, linear opening (b), closing (c)

Exercise n° 2

The purpose of this exercise is to get you used to the behavior of opening and closing. So, feel free to use the transformations generated in the previous exercise and apply them to the provided images (GRAINS1, GRAINS2, PARTIC1, PARTIC2, BALLS, METAL1, METAL2, SALT, KNITTING, MUSCLE). To help you in your quest, here are some guide-lines.

1) Verify the behavior of opening as a size distribution, by applying transformations of increasing size to the images GRAINS2, BALLS, SALT, KNITTING, MUSCLE.

2) The notion of size distribution (or granulometry) is not related to the notion of particle. Closings allow to perform (by duality) the size distribution of pores, and thus to reveal the spatial distribution of the connected components of a set (images PARTIC2, SALT, KNITTING, MUSCLE).

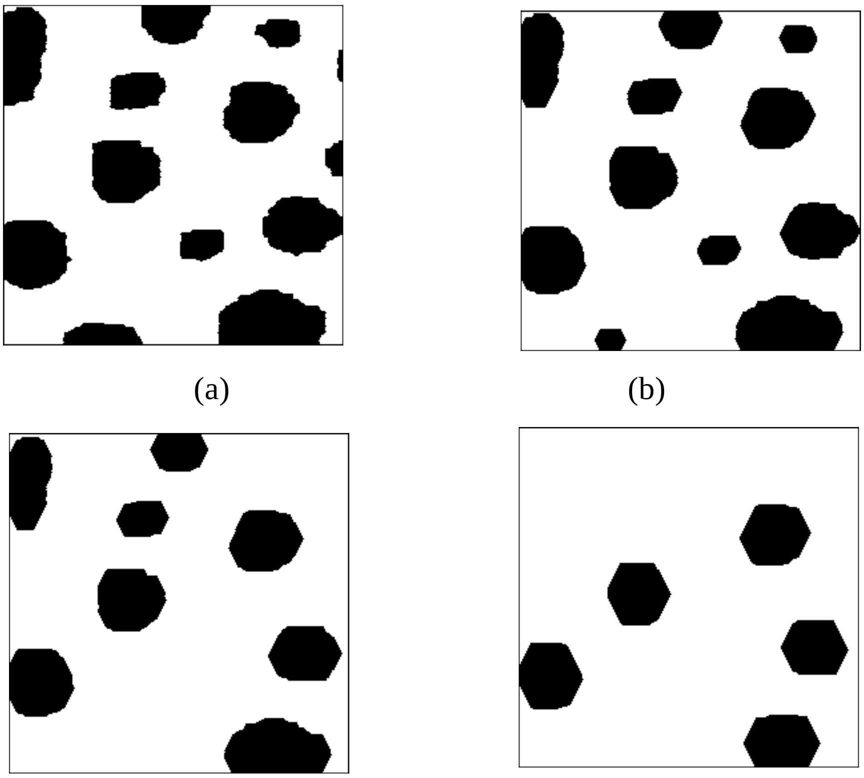
3) Both opening and closing "sieve" the connected components of a set according to their size, but also according to their shape. You can observe this by performing hexagonal openings of increasing size on the image BALLS. Besides, opening "hexagonalizes" the remaining connected components.

Note this selective effect on a greytone image by performing openings by segments of increasing size (images CIRCUIT and BURNER).

[procedures **granul** ; **isogranul** ; Ch. 11]

Solution

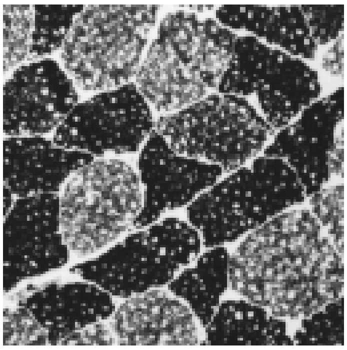
1) Opening by convex homothetics as a size distribution (or granulometry)



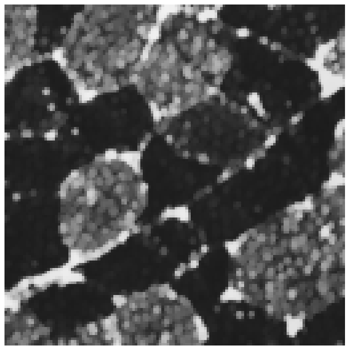
(a) (b) (c) (d)

Size distribution by opening

(a) initial image, (b) (c) (d), opening of increasing size



(a)



(b)

(c)

(d)

Size distribution by opening
(a) initial image, (b) (c) (d), opening of increasing size

The following images illustrate the use of the opening as a size distribution transformation. Transforms of increasing size act as a sieving device.

2) Size distribution by closing

The closing, which is the dual transformation with respect to the opening can be used to perform the size distribution of the space between particles. Thus, we fully use a property of the morphological notion of granulometry, which is to get rid of the concept of particle (or connected component).

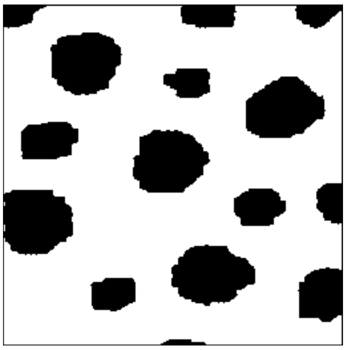
The procedure **granul** (in the file curves.mic) allows to compute the size distribution both on binary and greytone images.

3) Influence of the shape on the opening

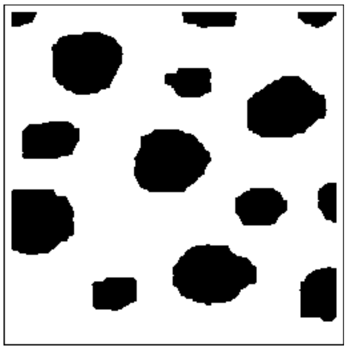
In order to verify the influence of the structuring element on the opening, it is useful to define the opening by dodecagons.

The procedures for binary and greytone images are given below.

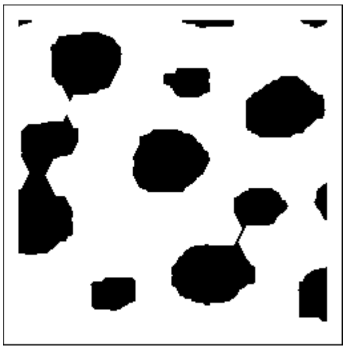
```
deproc ddcopen ddcopen s d sz
syntax "ddcopen binin binout size"
  int w ;
   w := imalloc imdepth s
   imcopy s d
   for 1 to sz do
   imcopy d w
   iminfngb 3 w d
   iminfngb 5 w d
```



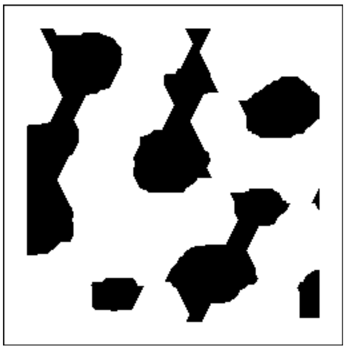
(a)



(b)

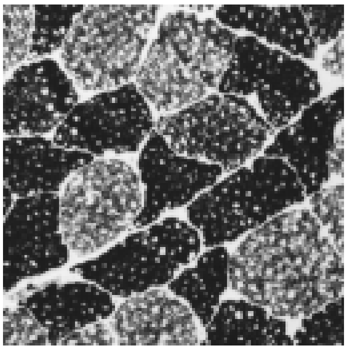


(c)

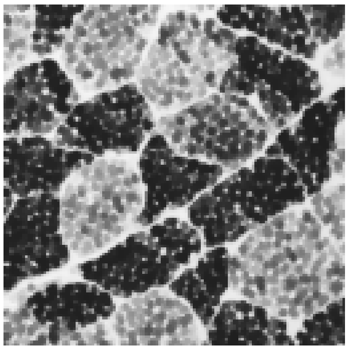


(d)

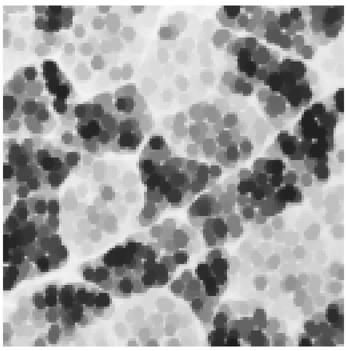
Size distribution by closing
(a) initial image, (b) (c) (d), closing of increasing size



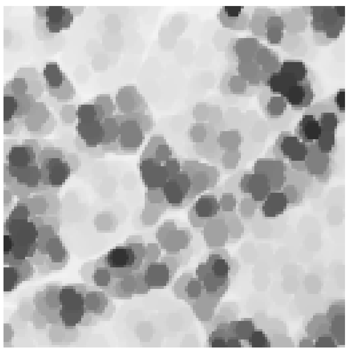
(a)



(b)



(c)

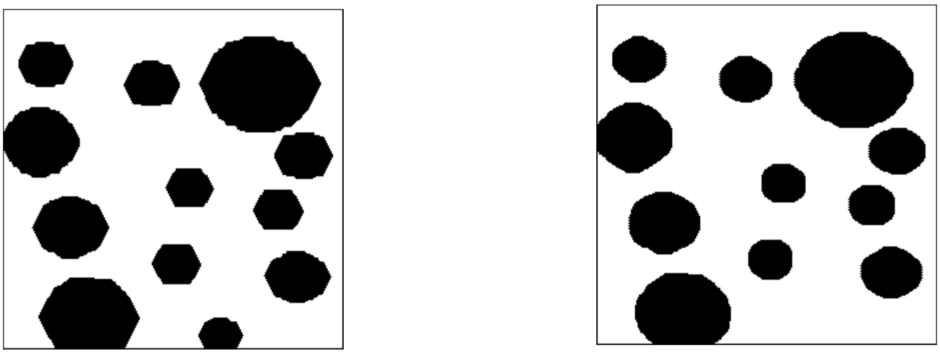


(d)

Size distribution by closing
(a) initial image, (b) (c) (d), closing of increasing size

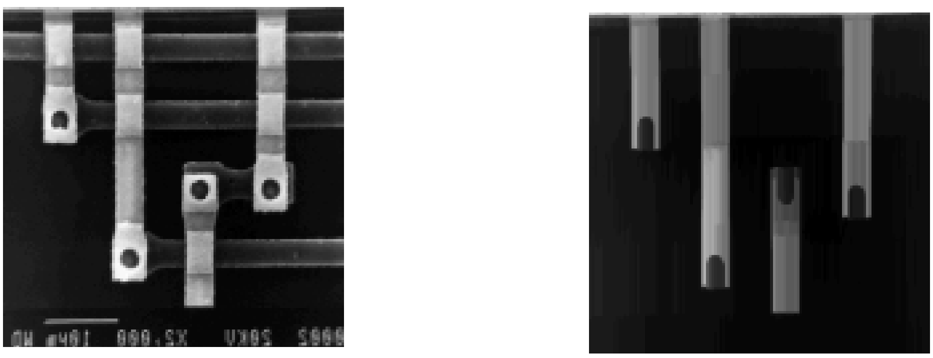
```
      iminfngb 1 w d
      imcopy d w
      iminfngb 4 w d
      iminfngb 6 w d
      iminfngb 2 w d
       ero d d 1
    end
    for 1 to sz do
      b1dil  d d
      b2dil  d d
      dil d d 1
    end
    imfree w
  end
```

If openings by vertical segments of great size are applied to the CIRCUIT image, we can see that only the tracks with a given orientation are preserved.



(a) (b)

(a) hexagonal opening
(b) dodecagonal opening of equivalent size



(a) (b)

(a) initial image
(b) opening by vertical segments of size 50

Exercise n° 3

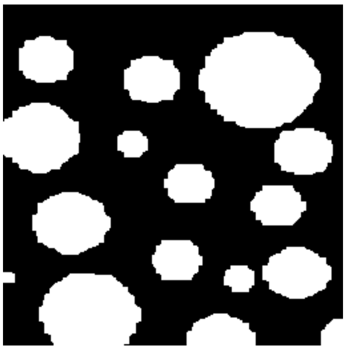
Opening and closing are good transformations for filtering noise in images. The NOISE image represents a set blurred by a scatter plot and the ELECTROP image represents a blurred greytone image.

1) Open the image with an elementary hexagon. Do the same with closing.

2) Perform the two operations successively. Does the order matter?

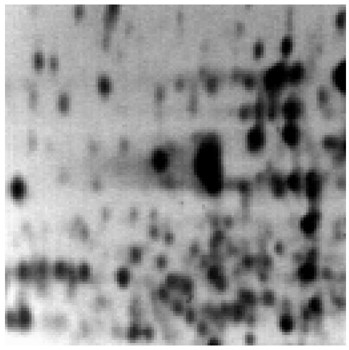
3) Can you enhance the filtering? (use triangular or 2x2 square openings and closings).

[procedures **open** ; **close** ; **miniopen** ; **miniclose**]



BALLS

NOISE

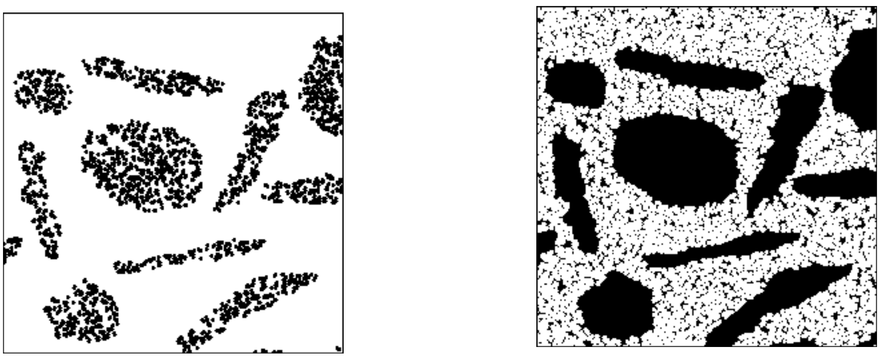


ELECTROP

Solution

1) Hexagonal opening and closing

The following images show the result of hexagonal opening and closing on the image NOISE.



hexagonal opening and closing of the image NOISE

2) Sequence of operations

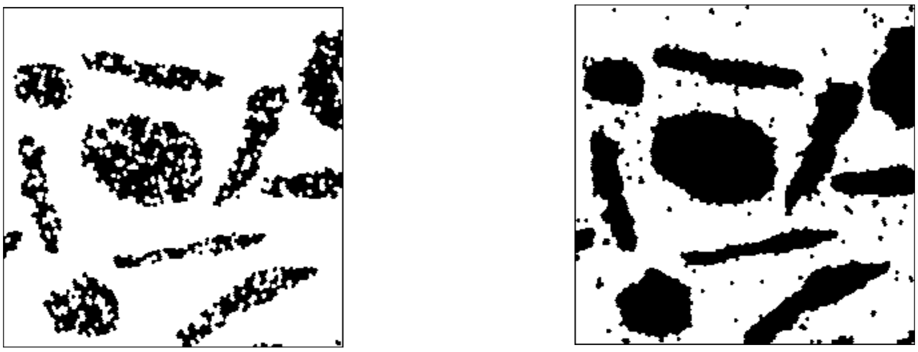
In the sequence of operations the order is important as shown by the two images below.

3) Using triangular or 2x2 square openings and closings:

```
deproc miniopen miniopen s d
syntax "miniopen binin binout"
  miniero s d
  minidil d d
```

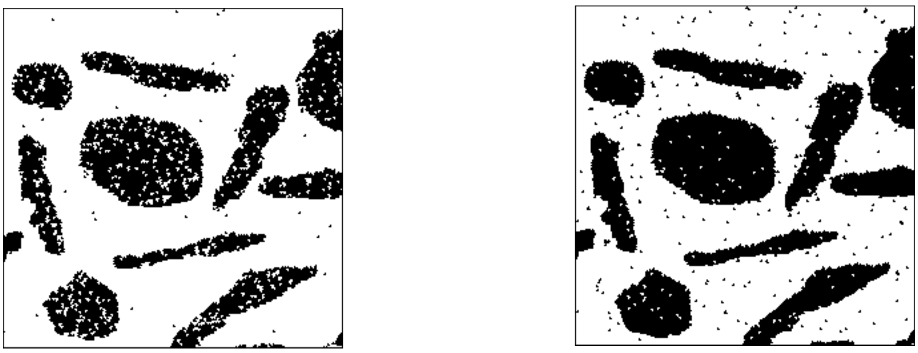
```
end

deproc miniclose miniclose s d
syntax "miniclose binin binout"
   minidil s d
  miniero d d
end
```

The order of operations still matters, yet to a lesser extent.



(a) (b)

(a) opening followed by hexagonal closing, (b) reverse order of operations



(a) (b)

(a) opening followed by triangular closing, (b) reverse order

In order to enhance filtering, the same operations should be performed with a reverse triangle.

Exercise n° 4 : Generalization of the notion of opening

The notion of opening can be defined in a more general way. Any transformation that is:

- increasing
- idempotent
- anti-extensive
- invariant under translation

The notion of closing is defined by duality: increasing, idempotent, extensive and invariant under translation.

1) Give some examples of openings and closings.

2) Is the operation consisting in extracting particles with one hole at least an opening? (binary case)

3) To obtain new openings, consider a family (γ_i) of openings. Prove that:

$\gamma = \sup_i \gamma_i$ resp. $\gamma = \bigcup_i \gamma_i$ is an opening.

$\varphi = \inf_i \varphi_i$ resp. $\varphi = \bigcap_i \varphi_i$ is a closing.

Program γ for the family of openings by segments in the three directions of the grid and the corresponding closing.

4) Observe on the CIRCUIT and TOOLS images that the new openings and closings have a different selective effect compared with those described up to now. We shall see further (chapter 5) how to construct other openings acting still differently.

[procedures **lineopen** ; **lineclose** ; **infopen** ; **supclose**]

Solution

1) The operation consisting in filling the holes is a closing. The operation consisting in extracting the particles whose surface is greater than N is an opening. On the other hand, the operation consisting in extracting the particles whose surface is smaller than N is not an opening (not increasing).

In numerical mode, the lopping of a function at level λ (all that is higher is reduced to value λ) is an opening (it is even a size distribution).

2) No, because the operation is not increasing. It suffices to take a connected component X that does not possess a hole and to take $Y \subset X$ equal to the interior contour of X.

3) Prove that $\sup_i \gamma_i$ is an opening :

Increasing:

Let f < g.

$\forall i, \gamma_i(f) \leq \gamma_i(g)$ for γ_i is an opening (hence increasing)

$\forall i, \gamma_i(f) \leq \sup_j \gamma_j(g)$

$\sup_i \gamma_i(f) \leq \sup_j \gamma_j(g)$ Q.E.D.

Anti-extensive :

$\forall i, \gamma_i(f) \leq f$

therefore $\left(\sup_i \gamma_i\right)(f) \leq f$ Q.E.D.

Idempotence :

Let us denote $\Psi = \sup_i \gamma_i$.

$\forall j, \Psi(f) \geq \gamma_i(f)$ by definition

$\forall j, \gamma_i(\Psi(f)) \geq \gamma_i(\gamma_j(f)) = \gamma_j(f)$

for γ_j is increasing and idempotent.

$\forall j, \left(\sup_k \gamma_k\right)(\Psi(f)) \geq \gamma_j(f)$ a fortiori.

then we have $\forall j, \Psi(\Psi(f)) \geq \gamma_j(f)$

then $\Psi(\Psi(f)) \geq \sup_j(\gamma_j(f))$

that is: $\Psi(\Psi(f)) \geq \Psi(f)$

hence $\Psi(\Psi(f)) = \Psi(f)$ since Ψ is anti-extensive. Q.E.D.

Translation invariant:

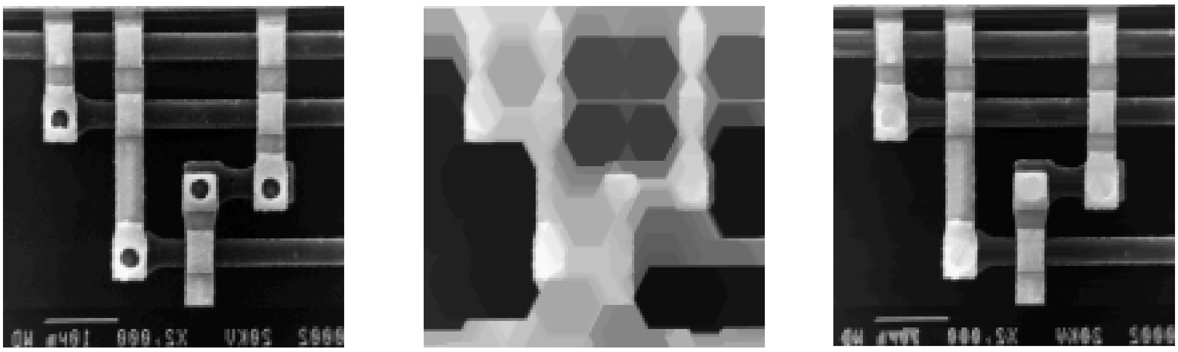
Obvious since so are the γ_i.

As an illustration, here are the procedures for defining the opening (resp. the closing) by means of linear opening (resp. closing) families using segments of various directions in the grid.

```
deproc lineopen lineopen s d sz
syntax "lineopen imin imout size"
  int w1 w2 i k;
   k := imdepth s
   w1 := imalloc k
   w2 := imalloc k
   imset impixmin w2 w2
   if (grid = 1) then
     i := 0
     for 1 to 3 do
       diropen ++i s w1 sz
        imsup w1 w2 w2
     end
   else
     i := 1
      for 1 to 2 do
         dirclose i s w1 sz
         iminf w1 w2 w2
         i := 3
      end
   end
   imcopy w2 d
   imfree w1
   imfree w2
 end

deproc lineclose lineclose s d sz
syntax "lineclose imin imout size"
    int w1 w2 i k;
    k := imdepth s
   w1 := imalloc k
   w2 := imalloc k
   imset impixmax w2 w2
   i := 1
   if (grid = 1) then
     for 1 to 3 do
       dirclose i s w1 sz
       iminf w1 w2 w2
       ++ i
     end
  else
     for 1 to 2 do
      dirclose i s w1 sz
      iminf w1 w2 w2
```

```
      i := 3
    end
  end
  imcopy w2 d
  imfree w1
  imfree w2
end
```

4) The following images show the effect of isotropic closing compared with closing by intersection of linear closings. The difference between the two is particularly obvious in the case of elongated objects. The first transformation filters the objects according to their thickness, and the second is controlled by their elongation: an elongated object is preserved even if it is narrow.



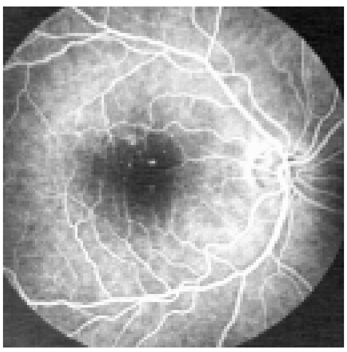
(a) (b) (c)

(a) initial image, (b) hexagonal closing of size 20
(c) family of linear closings of size 20

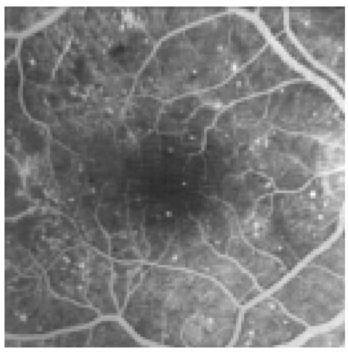
Exercise n° 5

1) Program the top-hat WTH(f) associated with the opening by a hexagon of size n and the conjugated top-hat BTH(f).

2) Apply these transformations to the CIRCUIT, ELECTROP, GRAINS3, RETINA1 and RETINA2 images.



RETINA1



RETINA2

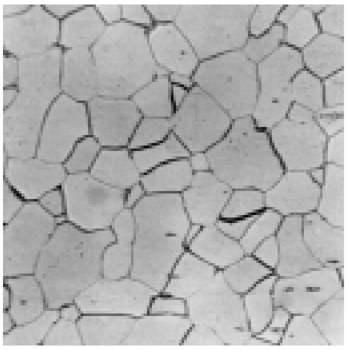
3) Note that these operations do not permit the discrimination of the aneurysm vessels (small white spots) on the images RETINA1 and RETINA2 images. Find a top-hat allowing such discrimination.

[procedures **openth** ; **closeth**]

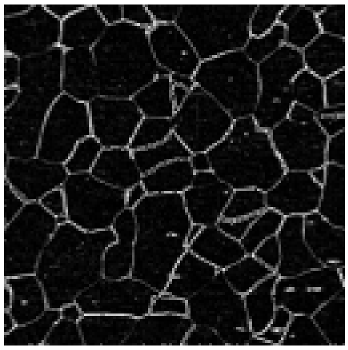
Solution

1) The hexagonal white top hat WTH(f) and the black top hat BTH(f) are obtained as follows:

```
deproc openth openth s d sz
syntax "openth greyin greyout size : white top hat"
   int w;
   w := imalloc imdepth s
   open s w sz
   imsub s w d
   imfree w
end

deproc closeth closeth s d sz
syntax "closeth greyin greyout size : black top hat"
   int w;
   w := imalloc imdepth s
   close s w sz
   imsub w s d
   imfree w
end
```

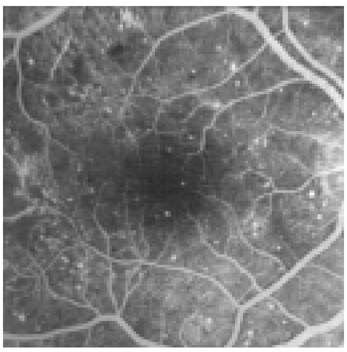


(a)

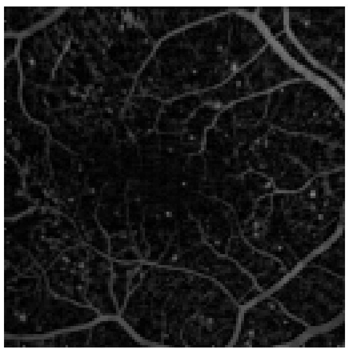


(b)

(a) initial image, (b) top hat by closing



(a)



(b)

(a) initial image, (b) top hat by opening

2) Application

The two examples above illustrate these transformations.

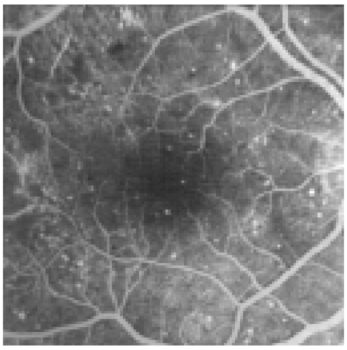
3) The top hat based on hexagonal opening does not allow to discriminate the aneurysm vessels. This operator extracts all that is narrower than a given size: whether the object is elongated or round, it disappears after the opening. The idea is then to use an opening that would make only the round objects (for example) disappear. Such a tool exists: the opening obtained as the sup of the family of the openings by segments.

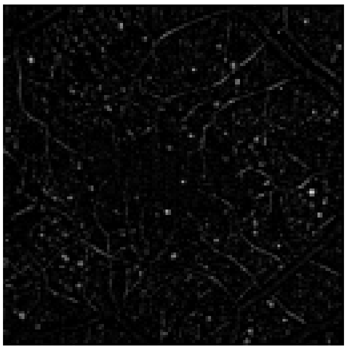
Then, the associated top hat can be constructed.

```
deproc lineopenth lineopenth s d sz
syntax "lineopenth greyin greyout size"
    int w;
   w := imalloc imdepth s
   lineopen s w sz
   imsub s w d
   imfree w
end
```

The dual operation may be defined as well :

```
deproc linecloseth linecloseth s d sz
syntax "linecloseth greyin greyout size"
   int w;
   w := imalloc imdepth s
   lineclose s w sz
   greysub s w d
   imfree w
 end
```





(a) (b)

(a) initial image, (b) top hat associated with openings by segments

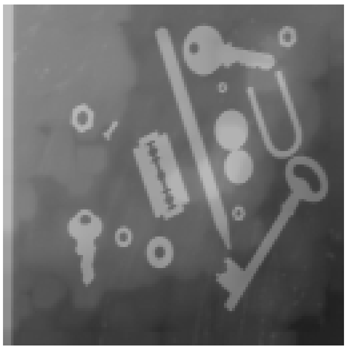
Exercise n° 6

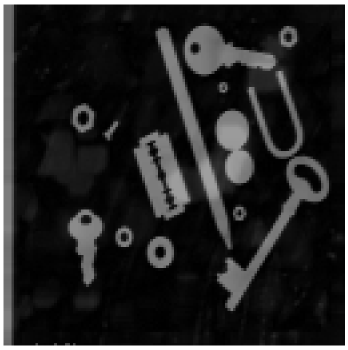
Demonstrate that a top-hat may serve to homogenize the grey level of the background of an image. Use the TOOLS image.

Solution

An opening of great size of the image TOOLS (the size must be greater than the object thickness) will suppress the particles while preserving the background. By subtracting the background, the top hat homogenizes the image. Then, it can be thresholded much easier. Even automated thresholding is often effective.

The following image shows the result of this top hat of great size. The value of the latter is not critical provided it is greater than the size of the objects. It could be determined automatically by analyzing the size distribution curve.


(a)


(b)

(a) initial image, (b) top hat by an opening of great size

SUMMARY

At the end of these exercises, your dictionary should be enriched with the following transformations:

open s d sz
hexagonal or square opening of size sz of the image s into the image d.
close s d sz
hexagonal or square closing of size sz of the image s into the image d.
diropen dir s d sz
opening by a segment of size sz in a direction dir of the image s into the image d.
dirclose dir s d sz
closing by a segment of size sz in a direction dir of the image s into the image d.
lineopen s d sz
sup of linear openings of size sz of the image s into the image d.
lineclose s d sz
inf of linear closings of size sz of the image s into the image d.
openth s d sz
top-hat associated with the opening by a hexagon of size sz of the greytone image s into the greytone image d.
closeth s d sz
top-hat associated with the closing by a hexagon of size sz of the greytone image s into the greytone image d.
lineopenth s d sz
top-hat associated with the family of openings by segments of size sz of the greytone image s into the greytone image d.
linecloseth s d sz
top-hat associated with the family of closings by segments of size sz of the greytone image s into the greytone image d.

Chapter 4

MORPHOLOGICAL FILTERS

4.1. Reminder

Before we can extract any object from a greytone image, it is often necessary to enhance the image. The enhancement of an image is mainly obtained by a filtering operation. A morphological filter is a transformation ϕ which has the two following properties:

(i) ϕ is increasing

(ii) ϕ is idempotent.

4.1.1. Sequential alternating filter

The white (resp. black) alternating sequential filter (ASF) consists, as the name indicates, in alternating morphological openings and closings (resp. closings and openings) of increasing size. Let γ_n be a size distribution and φ_n an anti-size distribution, the white alternating sequential filter of size n of a function f is defined by:

$$\phi_n(f) = \varphi_n \gamma_n \varphi_{n-1} \gamma_{n-1} \ldots \varphi_2 \gamma_2 \varphi_1 \gamma_1 (f)$$

Similarly, the black alternating sequential filter of size n of f is defined by:

$$\psi_n(f) = \gamma_n \varphi_n \gamma_{n-1} \varphi_{n-1} \ldots \gamma_2 \varphi_2 \gamma_1 \varphi_1 (f)$$

4.1.2. Morphological center

4.1.2.1. Definition

Let (Ψ_i) be a family of increasing transformations. Put:

$$\eta = \wedge \Psi_i \text{ et } \zeta = \vee \Psi_i$$

The center c of the family (Ψ_i) is:

$$c = (I \vee \eta) \wedge \zeta$$

(I represents the identity function).

The center is not a filter (it is not idempotent), but it is convergent when iterated and its limit is a filter.

4.1.2.2. Examples

1. for (Ψ_i) = ($\gamma\varphi$, $\varphi\gamma$) :

$$c(f) = [f \vee (\gamma\varphi(f) \wedge \varphi\gamma(f))] \wedge (\gamma\varphi(f) \vee \varphi\gamma(f))$$

2. for (Ψ_i) = ($\gamma\varphi\gamma$, $\varphi\gamma\varphi$) :

$$c(f) = [f \vee (\gamma\varphi\gamma(f) \wedge \varphi\gamma\varphi(f))] \wedge (\gamma\varphi\gamma(f) \vee \varphi\gamma\varphi(f))$$

This center is also called an automedian filter (its limit being a filter).

4.2. Contrasts, Reminder

Let η be an anti-extensive transformation and ξ an extensive transformation of a function f. A three-state contrast of primitives η and ξ is any transformation κ such that, for any f:

(i) $\kappa(f)(x)$ only depends on $\xi(f)(x)$, $\eta(f)(x)$ and f(x) and on possible constants.

(ii) $\kappa(f)(x)$ can only take one of these three values (the choice depending on a decision rule).

Besides, if κ(f)(x) cannot take the value f(x), the contrast is said to be a two-state contrast.

EXERCISES

Exercise n° 1

1) Without knowing it, you have already performed a sequential alternating filter (chapter 3, exercise n° 7) on the NOISE image by means of hexagonal and triangular openings and closings.

Continue this exercise:
- by increasing the number of iterations (size of the filter).
- by using different sizes of openings and closings (for example : a size of closing twice that of the opening).

2) Program the sequential alternating filters for greytone images by using the various openings and closings described up to now (hexagonal, triangular, by sup of linear openings, by erosion-reconstruction, etc.).

Test them on the RETINA3, BURNER and ELECTROP images.
[procedures **asf** ; **lineasf** ; **buildasf**]

Solution

1) By increasing the size of the alternating sequential filter, details, either white or black, of increasing size gradually disappear. The main objects and shapes are preserved while the contours are smoothed.

(a) (b) (c)

(d) (e) (f)

white alternating sequential filter of size (a) 1, (b) 4, (c) 3
black alternating sequential filter of size (d) 1, (e) 4, (f) 3

By using different sizes for openings and closings, the white pixels prevail over the black or vice-versa as the case may be.

2) ASF with various types of opening and closing:

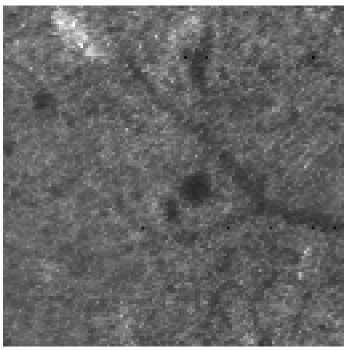
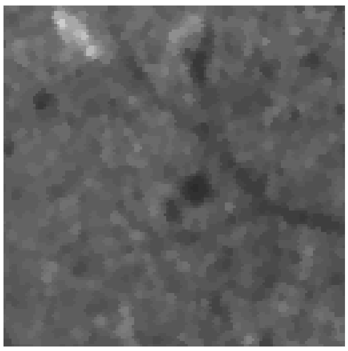
(a) opening and closing by hexagons

Let us define first the elementary ASF:

```
deproc af af s d sz t
 syntax "af imin imout size type"
    if (t <> 1) then
       open s d sz
       close d d sz
    else
      close s d sz
      open d d sz
    end
 end
```

Then the general ASF:

```
deproc fullasf fullasf s d sz t
syntax "fullasf imin imout size type"
   int i ;
   imcopy s d
   i := 1
   for 1 to sz do
     af d d i t
     ++ i
   end
 end
```

(a) (b)

(a) Original (detail), (b) white alternating sequential filter of size 1

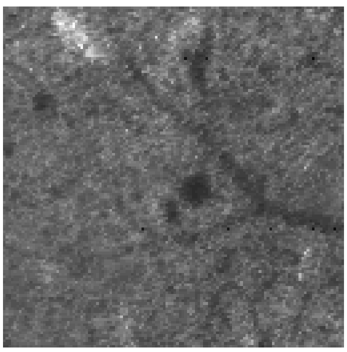
Note that the higher the sizes of the transformation, the longest the time of computation with no salient modifications appearing on the intermediary pictures. This is why a faster ASF can be defined where the size step is more important :

```
deproc asf asf s d sz  t
syntax "asf : imin imout size type"
```

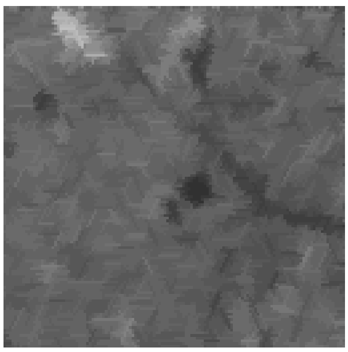
```
  int i j  ;
   i := 1  j := 0
   imcopy s d
   while (i < (sz + 1)) do
      af d d i t
      j := ( j + 1 )
      i := ( i + j )
   end
 end
```

(b) sup-opening of linear opening and inf-closing of linear closing

```
deproc lineaf lineaf s d sz t
syntax "lineaf imin imout size type"
   if (t <> 1) then
      lineopen s d sz
      lineclose d d sz
   else
     lineclose s d sz
     lineopen d d sz
  end
end

deproc lineasf lineasf s d sz  t
syntax "lineasf : imin imout size type"
   int i j  ;
   i := 1  j := 0
   imcopy s d
   while (i < (sz + 1)) do
      lineaf d d i t
      j := ( j + 1 )
      i := ( i + j )
   end
 end
```



(a)



(b)

(a) Original (detail), (b) linear white ASF of size 5

Exercise n° 2

Let κ be the contrast of primitives ξ and η and its decision rule the following:
if ξ(f)(x) - f(x) ≤ f(x) - η(f)(x)
then κ(f)(x) = ξ(f)(x) if not ξ(f)(x) = η(f)(x)

1) How many states is the contrast κ?

2) Program κ:

a) for η = erosion of size N and ξ = dilation of size N.
b) for η = morphological opening of size N and ξ = morphological closing of size N.
c) for η = opening of size N and ξ = closing of size 5*N.

Apply these transformations to the CHROMOSO image.

3) Verify that the contrast defined in 2-a is not idempotent and that the one defined in 2-b is idempotent.
[procedure **contrast**]

Solution

1) κ is a two-state contrast (obvious).

2) Programming κ

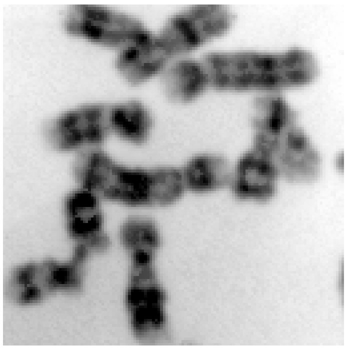
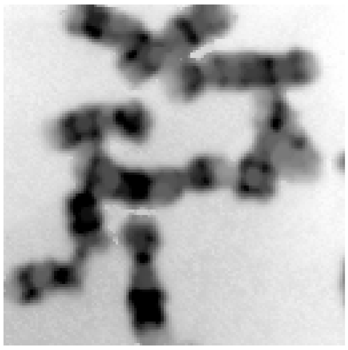
The procedure below uses one of the three combinations of transformations according to the value of the parameter t:

```
deproc contrast contrast s d sz t
 syntax "contrast greyin greyout size type (0: ero-dil, 1: open-close, 2: close-5*open)"
    int w1 w2 w3 w4 bw ;
    w1 := imalloc imdepth s
    w2 := imalloc imdepth s
    w3 := imalloc imdepth s
    w4 := imalloc imdepth s
    bw := imalloc 1
    if (t > 0) then
      if (t > 1) then
        open s w1 sz
        close s w2 (5 * sz)
      else
        open s w1 sz
        close s w2 sz
      end
    else
      ero s w1 sz
      dil s w2 sz
    end
    imsub s w1 w3
    imsub w2 s w4
    imsub w4 w3 w3
    imthresh w3 1 impixmax w3 bw
    immask bw 0 impixmax d d
    iminf w1 d d
```

```
 iminv bw bw
 immask bw 0 impixmax w4 w4
 iminf w2 w4 w4
 imsup w4 d d
 imfree w1
 imfree w2
imfree w3
imfree w4
imfree bw
end
```

The images below illustrate these various procedures.


(a) (b)

(c) (d)

Contrasts : (a) original image, (b) dilation/erosion contrast of size 1, (c) opening/closing contrast of size 6, (d) non symmetric opening/closing contrast of size 5

3) In order to validate or invalidate the idempotence of the different contrasts defined above, do:

```
contrast g1 g2 1 0
contrast g2 g3 1 0
imsub g3 g2 g4
print imvolume g4
```

Next:

```
contrast g1 g2 1 1
```

```
contrast g2 g3 1 1
imsub g2 g3 g4
print imvolume g4
```

Exercise n° 3

Let κ' be a transformation defined by:

$$\kappa'(f) = 3f - \gamma(f) - \varphi(f)$$

(γ is an opening, φ is a closing).

1) Is κ' a contrast in the sense given above?

2) Program κ' and apply it to the CHROMOSO image.

Solution

1) κ' may be written in a simpler way by remarking that:

$$\kappa'(f) = 3f - \gamma(f) - \varphi(f) = f + (f - \gamma(f)) - (\varphi(f) - f)$$

We recognize the white and black top-hats of f.

If, for all f, η(f) and ξ(f) are thus defined:

$$\eta(f) = \min(f, f + (f - \gamma(f)) - \varphi(f) - f)$$
$$\xi(f) = \max(f, f + (f - \gamma(f)) - (\varphi(f) - f))$$

we have, $\forall f$:

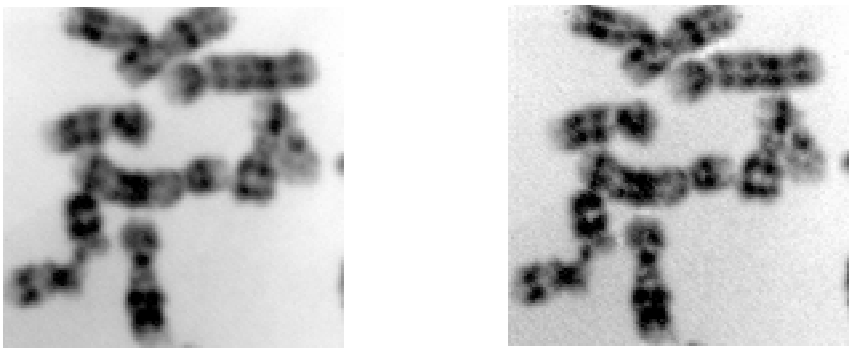
$$\eta(f) \leq f \text{ et } \xi(f) \geq f$$

then η is anti-extensive and ξ extensive.

κ' is identified with the contrast κ defined by:

$$\kappa(f)(x) = \eta(f)(x) \text{ if } \eta(f)(x) < f(x)$$
$$\kappa(f)(x) = \xi(f)(x) \text{ otherwise.}$$

2) The corresponding procedure is called Nthcontrast (see its definition below).



(a) (b)

Contrast by top-hat
(a) original, (b) transform of size 4

```
deproc contrasth contrasth s d sz
syntax "Nthcontrast greyin greyout size"
  int w1 ;
 w1 := imalloc imdepth s
 w := imalloc imdepth d
 imcopy s d
openth s w1 sz
```

```
  imadd w1 d d
 closeth s w1 sz
  imsub d w1 d
  imfree w1
end
```

Exercise n° 4

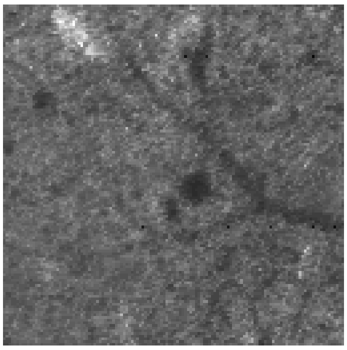
1) Program the automedian filter (which is not a filter according to the definition given above) and apply it to the RETINA3 and BURNER images.

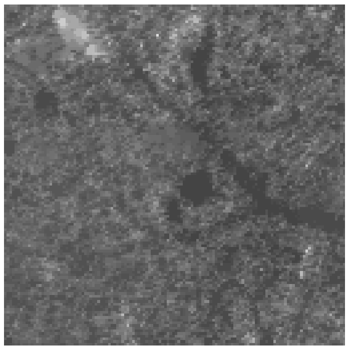
2) How many iterations are required for less than 10 % pixels to be modified? Try with different sizes.

Solution

1) Automedian filter

The procedure is the following:





(a) (b)

(a) Original (detail), (b) Automedian filter of size 5

```
deproc automed automed s d sz
 syntax "automed in out size"
    int oc co w sdepth;
    sdepth := imdepth s
    co := imalloc sdepth
    oc := imalloc sdepth
     w := imalloc sdepth
     af s oc sz 1
     af s co sz 0
     imcopy s d
     imcopy oc w
     imsup co w w
     iminf w d d
     imcopy oc w
     iminf co w w
     imsup w d d
     imfree w
     imfree co
     imfree oc
```

```
end
```

2) In order to compare the images and compute the number of modified points, we can use the **masksup** procedure given below. The following procedure is defined in which each time a key other than <Enter> is hit, a new iteration is started with the display of the number of modified points.

```
 deproc masksup masksup g1 g2 b
 syntax "masksup greyin1 greyin2 mask_out"
   int w ;
   w := imalloc imdepth g1
   imdiff g1 g2 w
   imthresh w (impixmin g1 +1) impixmax g1 b
   imfree w
end

deproc iterautomed iterautomed s d sz
  syntax "iterautomed greyin greyout size"
    int w b1 b2 nb v ;
     b1 := imalloc 1
     b2 := imalloc 1
     w := imalloc imdepth s
     imcopy s w
     v := readkey
     while (v <> 13) do
        automed w d sz
        immasksup w d b1
        masksup d w b2
        imor b1 b2
        nb := imvolume b2
       print [ "number of modified pixels: " nb ]
        imcopy d w
        v := readkey
    end
    imfree b1
    imfree b2
    imfree w
end
```

Thus, with a filter of size 1, 40852 pixels of the RETINA2 image are modified at the first iteration, and none at the second. Whereas the number of modified pixels in the BURNER image is 32449 at the first iteration, 1230 at the second, 326 at the third and 91 at the fourth.

SUMMARY

You are advised to keep the following transformations in your dictionary:

af s d sz type
sequential alternating filter of size sz of the image s into the image d, starting with **close** followed by **open** (type 1) or with **open** followed by **close** (type 2).
fullasf s d sz type
sequential alternating filter of size sz of the greytone image s into the greytone image d, starting with the closing **close** of size 1 if type=1 or with the opening **open** of size 1 if type is not equal to 1.
lineasf s d sz type
sequential alternating filter of size sz of the greytone image s into the greytone image d, starting with the closing **lineclose** of size 1 if type=1 or with the opening **lineopen** of size 1 if type is not equal to 1.
buildasf s d sz type
sequential alternating filter of size sz of the greytone image s into the greytone image d, starting with the closing **buildclose** of size 1 if type=1 or with the opening **buildopen** of size 1 if type is not equal to 1.
contrast s d sz type
contrast of size sz of the greytone image s into the greytone image d by erosion and dilation if type=1, by opening and closing if type=2, by opening and closing, the size of the closing being five times larger than that of the opening if type>2.
contrasth s d sz
contrast by white and black top-hat of size sz of the greytone image s into the greytone image d.
automed s d sz
morphological center between the two alternating filters **af** (sz), type 0 and 1.

Chapter 5

GEODESY AND CONNECTIVITY

5.1. Reminder, binary case

Given a set X, the geodesic distance between two points x and y of X is defined as the length of the shortest path L(x,y) included in X and joining these two points. We can now define balls of size λ in this metric system, and then the erosion and dilation of a set Y included in X by a geodesic ball B .

When one works on digitized sets, it can be shown that the elementary geodesic dilation is defined by:

$$D_X(Y) = (Y \oplus H) \cap X$$

Similarly, the elementary geodesic erosion is defined by:

$$E_X(Y) = X \cap [(Y \cup X^c) \ominus H]$$

H being an elementary digital ball (hexagon or square).

5.2. Greytone case

In the greytone case, the geodesic space under consideration may be either a set X or a function g (the latter case is the immediate generalization of the binary notion applied to sub-graphs). On digitized sets, the following definitions apply:

The geodesic dilation of f into the set X by an elementary hexagon centered in O is defined at any point x by:

$$D_X(f)(x) = \sup_{\substack{b \in H \\ x+\overrightarrow{Ob} \in X}} \left(f\left(x + \overrightarrow{Ob} \right) \right) = \sup\{ f(x); x \in H_x \cap X \}$$

Likewise the erosion:

$$E_X(f)(x) = \inf_{\substack{b \in H \\ x+\overrightarrow{Ob} \in X}} \left(f\left(x + \overrightarrow{Ob} \right) \right) = \inf\{ f(x); x \in H_x \cap X \}$$

The geodesic dilation of f conditionally to the function g by an elementary hexagon centered in O is defined by:

$$D_g(f) = \inf(f \oplus H, g)$$

Likewise the erosion:

$$E_g(f) = \sup(f \ominus H, g)$$

NB : Note that the duality is different from the one defined in the binary case. Here we simply replace f by m-f (where m=255 for 8 bits images).

EXERCISES

Exercise n° 1

The purpose of the first exercise is to program the geodesic erosions and dilations of size n defined above. Make sure your algorithm is elaborated so that the transformation is accurate into the special case where X equals D, the field of measurement.
[procedures **gdsdil** ; **gdsero**]

Solution

To program a geodesic erosion which is exact even when the field X is equal to D, the parameter **edge** must be equal to 1.

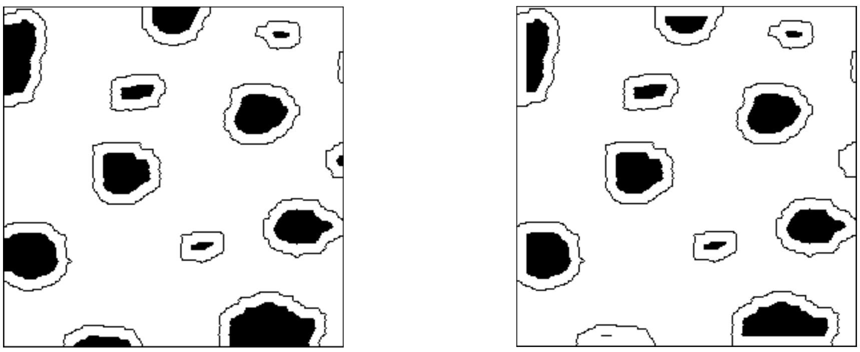
The definition of these transformations can be the following (m is the geodesic set, s the initial set):

```
deproc gdsdil gdsdil s m d sz
syntax "gdsdil in bin_or_grey_mask out size"
  int z w;
  z := imalloc imdepth d
  imcopy s z
  if ((imdepth m > 1) | (imdepth s = 1 )) then
    for 1 to sz do
      dil z z 1
      iminf m z z
    end
  else
    w := imalloc imdepth s
    immask m 0 (impixmax w) w
    iminf w z z
    for 1 to sz do
      dil z z 1
      iminf w z z
    end
    imfree w
  end
  imcopy z d
  imfree z
end
```

The erosion can be obtained by duality (dilation of the complementary set). The procedure is split into two parts, a binary one and a greytone one:

```
deproc bingdsero bingdsero s m d sz
syntax "bingdsero binin binmask binout size"
  int w ;
  w := imalloc imdepth d
  imcopy s w
  for 1 to sz do
```

```
      imdiff m w w
      dil w w 1
      imdiff m w w
    end
    imcopy w d
    imfree w
end

deproc greygdsero greygdsero s m d sz
syntax "greygdsero greyin bin_or_grey_mask greyout size"
    int w1 w2 ;
    w1 := imalloc imdepth s
    w2 := imalloc imdepth m
    iminv s w1
    iminv m w2
    gdsdil w1 w2 d sz
    iminv d d
    imfree w1
    imfree w2
end

 deproc gdsero gdsero s m d sz
 syntax " gdsero bin_or_greyin bin_or_grey_mask bin_or_greyout size"
    if (imdepth s > 1) then
      greygdsero s m d sz
    else
      bingdsero s m d sz
    end
 end
```



(a) (b)

Geodesic and Euclidean erosions
(a) geodesic erosion (the mask is constituted with the field of analysis), (b) Euclidean erosion

Exercise n° 2 : Binary reconstruction

Let X be a set composed of several connected components $\{X_i\}$. A set Y included in X marks one or several connected components of X. The present exercise consists in reconstructing the connected components of X marked by Y.

These connected components consist of points x of X which are at a finite geodesic distance d(x,Y) from Y.

Define the reconstruction algorithm by means of geodesic dilations. What will you use as ending criterion?
[procedure **build**]

Solution

ms is the initial set, mq is the marking set as well as the set containing the result of reconstruction.

```
deproc binbuild binbuild ms mq
syntax "binbuild binmask binmarker"
  int a b ;
  b := imvolume mq
  while (a <> b) do
    a := b
   gdsdil mq ms mq 1
    b := imvolume mq
  end
end
```

We know that reconstruction is achieved when the surface of the reconstructed set ceases to increase.

Exercise n° 3 : Greytone reconstruction

The **build** function, which is the geodesic dilation of f conditionally to g until idempotence, is already efficiently installed in MICROMORPH.

1) Test this function on the TOOLS image by using as a marker an image entirely set to 0 except in a point (selected preferably at the location of an object) where the value is set to 255.

2) Do the same operation on the RETINA1 image, by placing the point on the blood lattice.
[procedure **build**]

Solution

1) Greytone reconstruction

After the TOOLS image is loaded in g1, we proceed as follows:

```
imset 0 g2
imwritepix 255 118 155 g2
build g1 g2
```

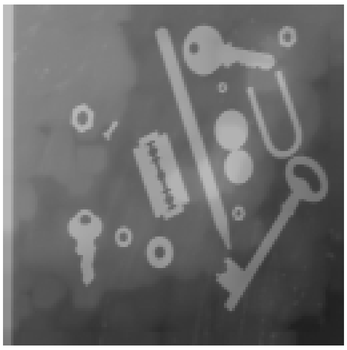
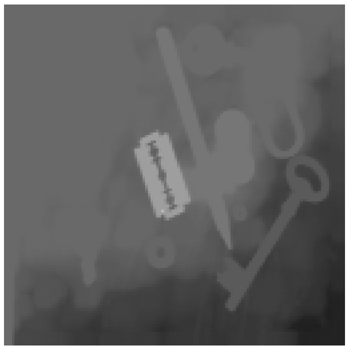
Here, the behavior is comparable to the binary reconstruction, only the marked object is reconstructed.

2) The same effect can be noticed on the image RETINA1. (Load the image RETINA1 in g1).

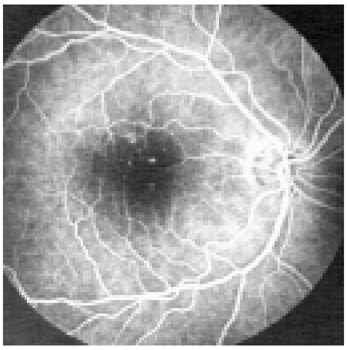
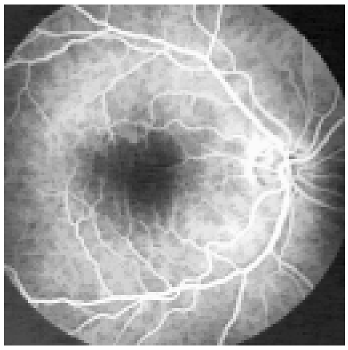
```
imset 0 g2
```

imwritepix 255 210 145 g2
build g1 g2

Here, we notice that the white spots situated at the center are not reconstructed. Indeed, they are not connected to the network. x and y are connected will mean: there exists a downward way joining a point x of the marker to a point y of the mask.

(a) (b)
(a) original, (b) result of greytone reconstruction

(a) (b)
(a) original, (b) result of greytone reconstruction

Exercise n° 4: Opening by erosion - reconstruction

Prove that the reconstruction by geodesic dilation of the erosion of f (resp. X) of size n by the structuring element B conditionally to f (resp. X) is an opening (the center of the structuring element B must be a point of B).

Program these operations, as well as the dual closings and compare them on the RETINA1, RETINA2 and CAT images with the openings previously described.
[procedures **buildopen** ; **buildclose**]

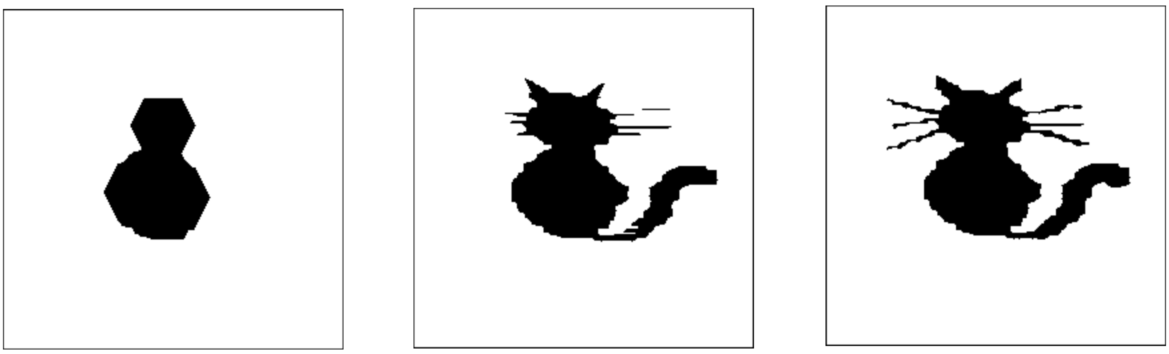
Solution

Denote $\Psi_0(X) = X \ominus nH$, $\Psi_m(X) = (\Psi_{m-1}(X) \oplus H) \cap X$.
Ψ designates the limit of Ψ_m when $m \to \infty$.
<u>Increasing</u>:
Let $Y \subset X$. $Y \ominus nH \subset X \ominus nH$ since erosion is increasing.
$(Y \ominus nH) \oplus H \subset (X \ominus nH) \oplus H$ since dilation is increasing.
$((Y \ominus nH) \oplus H) \cap Y \subset ((Y \ominus nH) \oplus H) \cap X \subset ((X \ominus nH) \oplus H) \cap X$ since $Y \subset X$.
Then $\Psi_1(Y) \subset \Psi_1(X)$. By iteration, $\Psi_m(Y) \subset \Psi_m(X)$. If $\Psi(X)$ designates the limit of this iteration when $m \to \infty$, we have $\Psi(Y) \subset \Psi(X)$.

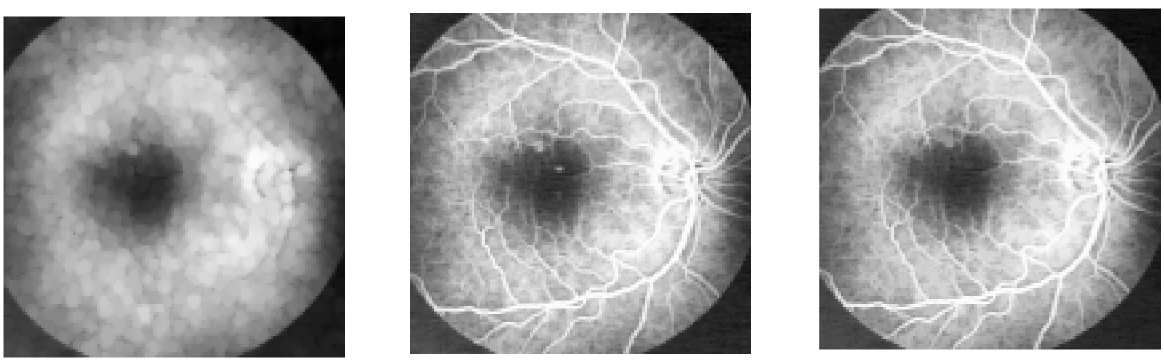
<u>anti-extensive</u>:
obvious.
<u>idempotent</u>:
In order to prove that $\Psi(\Psi(X)) = \Psi(X)$, we will prove that $X \ominus nH \subset \Psi(X) \ominus nH$. From we can immediately deduce that $\Psi(\Psi(X)) = \Psi(X)$, since Ψ is anti-extensive and the erosion is increasing ($\Psi(X) \ominus nH \subset X \ominus nH$).
We still have to show that $X \ominus nH \subset \Psi(X) \ominus nH$.
Let us prove first that:
1. $\Psi_m(X) \subset \Psi(X)$
2. $\forall m \leq n, \Psi_m(X) = (X \ominus nH) \oplus mH$
3. $((X \ominus nH) \oplus nH) \ominus nH = X \ominus nH$
1. As the center of H is taken in H, then $\forall m, \Psi_m(X) \subset \Psi_{m+1}(X)$. At the limit we have $\Psi_m(X) \subset \Psi(X)$.
2. Let $m \leq n$. We have:

$$(X \ominus nH) \oplus mH \subset (X \ominus nH) \oplus nH \subset X$$

Then $\Psi_m(X) = (X \ominus nH) \oplus mH$, since the intersection with X at each step does nothing.
3. $\subset$: because the opening is anti-extensive.
$\supset$: because the closing is extensive.
Now consider m>n. $\Psi n(X) \subset \Psi_m(X)$, then:
$\Psi_n(X) \ominus nH \subset \Psi_m(X) \ominus nH$ et $\Psi_n(X) \ominus nH = ((X \ominus nH) \oplus nH) \ominus nH$ (2nd property).
And also (3rd property) : $\Psi_n(X) \ominus nH = X \ominus nH$.
Then, $\forall m > n, X \ominus nH \subset \Psi_m(X) \ominus nH$.
At the limit, we have $X \ominus nH \subset \Psi(X) \ominus nH$ Q.E.D.



(a) (b) (c)
(a) Hexagonal opening, (b) Sup of linear openings, (c) Opening by erosion-reconstruction



(a) (b) (c)
(a) Hexagonal opening, (b) Sup of linear openings, (c) Opening by erosion-reconstruction

The transformations are the following:

```
 deproc buildopen buildopen s d sz
 syntax "buildopen binin binout size"
   int w ;
   w := imalloc imdepth s
   ero s w sz
   build s w
   imcopy w d
   imfree w
end

deproc buildclose buildclose s d sz
syntax "bbuildclose binin binout size"
   int w ;
   w := imalloc imdepth d
   iminv s s
   buildopen s w sz
   iminv s s
   iminv w d
   imfree w
end
```

SUMMARY

This chapter of exercises should have allowed you to introduce the following transformations into the dictionary :

gdsdil s m d sz
geodesic dilation of size sz of the image s into the image d.
gdsero s m d sz
geodesic erosion of size sz of the image s into the image d.
build m imout
geodesic reconstruction of the image imout into the image m.
buildopen s d sz
reconstruction by geodesic dilation of the erosion of size sz of the image s into the image d.
buildclose s d sz
dual transformation.

Chapter 6

APPLICATIONS OF GEODESY

The first application of geodesy is (binary and greytone) reconstruction, which was presented in the preceding chapter. This leads to introduce other applications, as many of them use reconstruction.

EXERCISES

Exercise n° 1: Individual analysis of particles

1) Design an algorithm allowing the individual analysis of particles, in other words capable of extracting every connected component from an image in order to measure it.

2) Application to the measurement of the area of grains on the PARTIC1 image.

3) Verify that the transformation $\Psi(X)$ defined in this way:

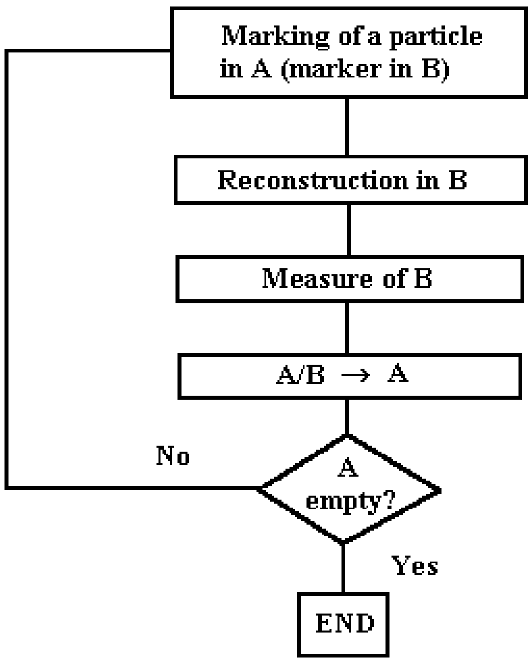
$$X = \bigcup_{i=1}^{n} X_i,\ X_i \text{ connected component of X}$$
$$\Psi_\lambda(X) = \cup X_j \text{ such that } \mathrm{mes}(X_j) > \lambda$$

is a size distribution.
[procedure **fgrain**]

Solution

1) Individual analysis of particles

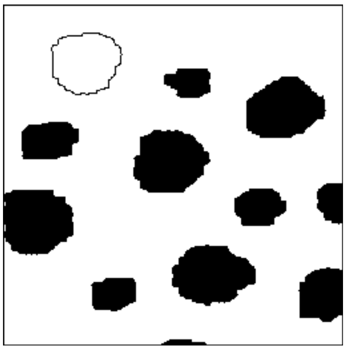
Individual analysis of particles consists in marking and reconstructing successively each particle of a set. This done, a certain number of measurements can be performed on the



particle before it is suppressed and the next particle is examined, and so on, until all particles of the initial set are eliminated.

The flow chart of such an algorithm is given above. The equivalent MICROMORPH procedure applied to the measurement of the area will be described later on.

2) Measuring the area of the grains of PARTIC1



Reconstruction and elimination of a particle

```
syntax "fgrain binin binout:in d, first particle of s "
   int w flag ;
   w := imalloc imdepth d
   imset 0 w
   flag := imcompare s w w
   if (flag <> -1) then
     build s w
     imdiff s w s
   end
   imcopy w d
   imfree w
end
```

The marking is performed by the **imcompare** transformation which extracts the first point of the first particle encountered in the image.

To start this operation on the image PARTIC1, proceed to the following operation, after the image has been loaded in b1:

```
fgrain b1 b2
print imvolume b2
```

The operation performed on every particle of the image gives you the table of results below.

3) $\Psi_\lambda(X)$ is a size distribution (granulometry):

- It is idempotent : indeed, to keep of a set only the connected components whose surface is higher than It is idempotent : indeed, to keep of a set only the connected components whose surface is higher than λ, then repeat the operation with the new set, the result remains unchanged.
- When $\lambda \geq \mu$, $\Psi_\lambda(X) \subset \Psi_\mu(X)$ (obvious).
- Finally: $\Psi_\lambda\left[\Psi_\mu(X)\right] = \Psi\mu[\Psi\lambda(X)] = \Psi_{\sup(\lambda,\mu)}(X)$.

N° particle	Area
1	1 408
2	1 658
3	1 455
4	11 983
5	1 620
6	12 068
7	11 038
8	12 109
9	1 517
10	1 876
11	12 150
12	12 093
13	11 170
14	1 720
15	1 144

Exercise n° 2: Holes filling (binary and greytone), objects cutting edges

1) Apply the geodesic reconstruction algorithm to suppressing the particles cutting the field border. What can be in particular the marking set Y? Application to the GRAINS2 image.

2) How can you fill the holes in the particles? Design an algorithm and test it on the HOLES and GRUYERE images.

3) On a greytone image a hole may correspond to what is called a basin if the image is considered as a relief. We can then imagine to fill up these holes, as would rain water, the exceeding water spilling outside the limits of the image. Similarly to the binary case, design an algorithm for filling the holes on a greytone image and apply it to the CIRCUIT and TOOLS images.

[procedure **cloholes**]

Solution

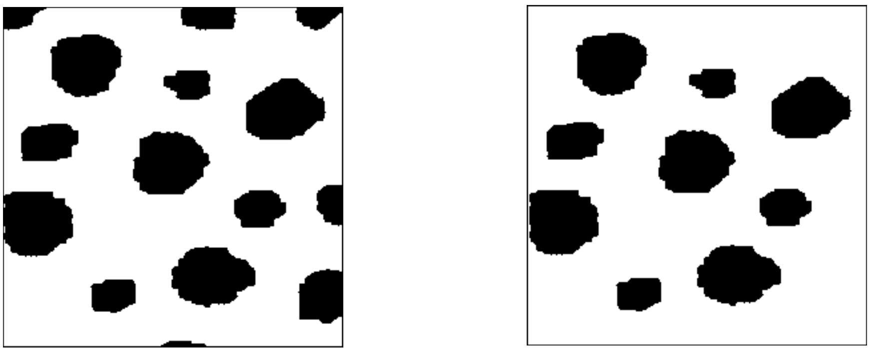
1) Suppressing the particles which cut the field edges

The marking set consists of the intersection of the initial set s and the field contour.

```
deproc  border border z
syntax "border binout (z = border of the bin-image field)"
  int k ;
  k := edge
  imsetedge 0
 imset impixmax z z
 iminfngb z z 1 1
 iminfngb z z 3 1
 iminfngb z z 5 1
 if ( grid = 0 ) then
    iminfngb z z 7 1
  end
  iminv z z
  imsetedge k
```

```
end

deproc edgeoff edgeoff s d
syntax "edgeoff binin binout"
  int w ;
  w := imalloc imdepth d
  border w
  iminf s w w
  build s w
  imdiff s w w
  imcopy w d
  imfree w
end
```

(a) (b)



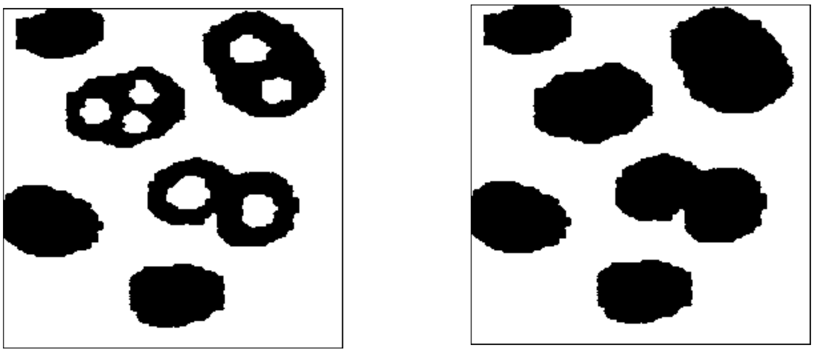
Suppression of the objects which cut the field edges
(a) initial set, (b) result of transformation

2) Closure of the holes (binary image)

The initial set for the geodesic reconstruction is the complemented set. Then, the holes become the connected components which cannot be accessed from the field edge. The marker is the same as in the preceding example :

```
deproc clohole clohole s d
syntax "clohole binin binout"
  int w ;
  w := imalloc imdepth s
  iminv s s
  border w
  iminf s w w
  build s w
  iminv s s
  iminv w d
  imfree w
end
```

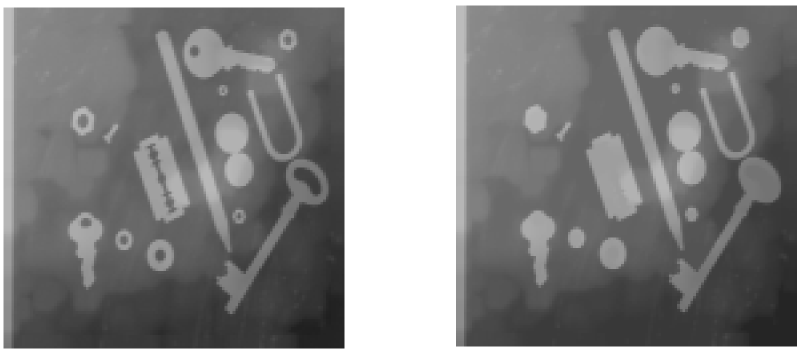
(a) (b)



Closure of the holes, (a) initial set, (b) objects without holes

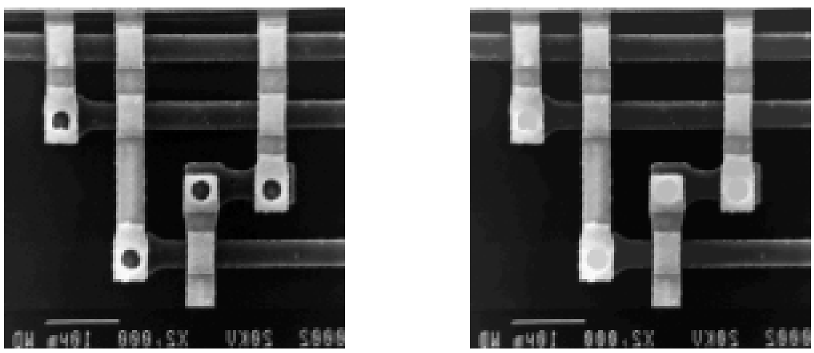
3) Closure of the holes (greytone image)

(a) (b)



Closure of the holes, (a) initial set, (b) objects without holes

(a) (b)



Closure of the holes, (a) initial set, (b) objects without holes

Only the reconstruction by dilation is installed in MICROMORPH. The image has to be inverted. It will be our mask. The used marker is an image with the value 255 on the borders and 0 everywhere else.

Exercise n° 3 :Regional maxima and minima (greytone case)

Let p_0 ,...,p_n be the points of the image I. $V(p_i)$ is the value of the image in p_i .

$p_0\ p_1 \ldots p_n$ is called a path if, $\forall i$, p_{i+1} et p_i are neighbors.

$p_0 \ldots p_n$ is said to be strictly ascending if, $\forall i$, $V(p_{i+1}) \geq V(p_i)$ et $V(p_0) < V(p_n)$.

$p_0 \ldots p_n$ is said to be strictly descending if, $V(p_{i+1}) \leq V(p_i)$ et $V(p_0) > V(p_n)$

A connected set X is a regional maximum if there exists no strictly ascending path coming from X:

$$\forall x \in X, \forall (p_i)_{i\in\langle 1...N\rangle} \in I^N, xp_0...p_N \text{ is not strictly ascending.}$$

1) Find an algorithm allowing to determine the regional maxima using the successive thresholds of the image and binary reconstruction.

2) In practice, one does not use this method but the following: 1 is subtracted from the image and the resulting image is reconstructed conditionally to the initial image. The regional maxima are located where the resulting image differs from the initial image. The dual algorithm generates the regional minima.

Program this transformation. Apply it to the ELECTROP image. What do you observe? How can you explain this? Can you propose a solution?

3) The algorithmic method above allows to generalize the notions of minima and regional maxima. The extended regional maxima at a height h are obtained by subtracting the constant h from the initial image. The extended regional maxima are located where the resulting image differs from the initial image. The dual algorithm generates the extended regional minima.

Program this transformation. Apply it for increasing heights to the ELECTROP image. [procedures **maxima** ; **minima** ; **extmaxima** ; **extminima**]

Solution

1) Denote S_n the threshold between n and 255 of the image f:

$S_n(x) = 1$ if $f(x) \geq n$

$S_n(x) = 0$ if $f(x) < n$.

First, prove the following lemma:

Lemma: if S_n contains a connected component C that is not found in S_{n+1} , then C is a regional maximum.

<u>Proof</u>: $\forall x \in C$, f(x)=n since f(x)<n+1 $(x \notin S_{n+1})$ and f(x)≥n $(x \in S_n)$. C being a connected component of S_n , for any neighbor of a point of C, $S_n(y)=0$, then f(y)<n. Consequently, there cannot exist a strictly ascending path starting from a point of C.

Then it suffices for each threshold S_n, to reconstruct the connected components marked by the threshold S_{n+1} and to keep the connected components that have not been reconstructed.

Hence the algorithm:

```
deproc threshmax threshmax s d
syntax "threshmax greyin binout -> maxima successive thresholds"
  int level lev1 sn snplus1;
  sn := imalloc 1
  snplus1 := imalloc 1
  imset 0 d
  level := impixmax s
  lev1 := level
  imthresh s level level sn
  for 1 to lev1 do
    -- level
    imcopy sn snplus1
    imthresh s level lev1 sn
    binbuild sn snplus1
    imdiff sn snplus1 snplus1
```

```
    imor snplus1 d d
  end
  imfree sn
  imfree snplus1
end
```

It is a costly method as all the thresholds of the image have to be determined.

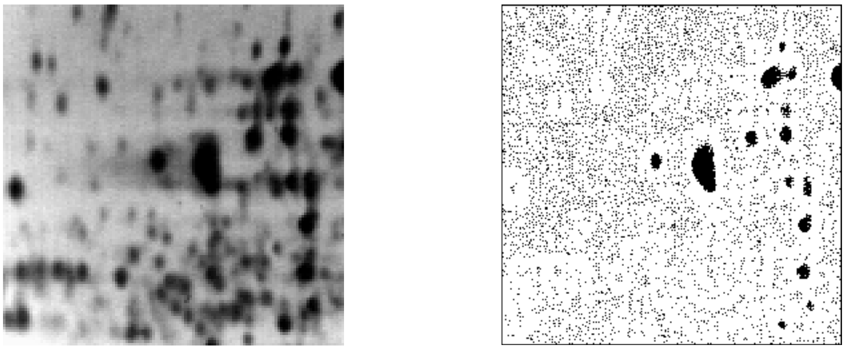
2) Algorithm by direct reconstruction

The algorithm by direct reconstruction uses the **build** procedure installed in MICROMORPH.

```
deproc maxima maxima s d
syntax "maxima greyin binout"
  int w;
  w := imalloc imdepth s
  imcopy s w
  imcsub w 1 w
  build s w
  imsub s w w
  imthresh w 1 impixmax s d
  imfree w
end
```

(a) (b)

Minima by direct reconstruction, (a) original image, (b) minima

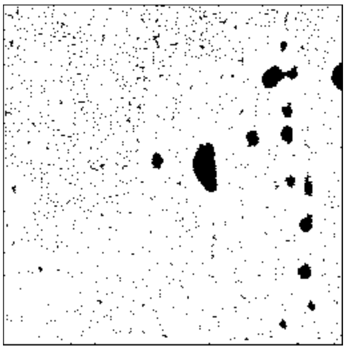


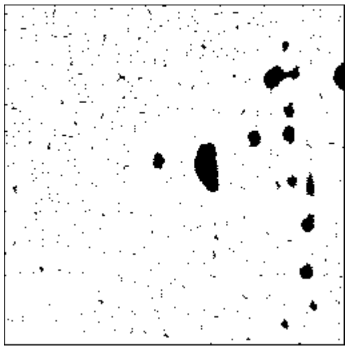
```
deproc minima minima s d
syntax "minima greyin binout"
  iminv s s
  maxima s d
  iminv s s
end
```

As can be seen, the electrophoresis image contains a great number of regional minima, that are most often reduced to one point. This is typical of a noisy image. Then, we may apply the filters described above in order to suppress, partly at least, the noise from the image. For example, perform a hexagonal opening and determine the new minima. The result of such an operation is illustrated on the following images.

(a) (b)

Minima of the filtered image
minima of the opened image (a) of size 1, (b) of size 2





3) Determining the extended maxima

These extended maxima are characterized by the fact that they are not only maxima, but they also are at a height h, which means that it is necessary to go down from a level higher than h before one is able to follow another ascending path towards another maximum. The procedure is the following:

```
deproc extmaxima extmaxima s d h
syntax "extmaxima greyin binout hauteur"
   int w;
   w := imalloc imdepth s
   imcopy s w
   imcsub w h w
   build s w
   imsub s w w
   imthresh w 1 impixmax s d
   imfree w
end

deproc extminima extminima s d h
syntax "extminima greyin binout hauteur"
   iminv s s
   extmaxima s d h
   iminv s s
end
```

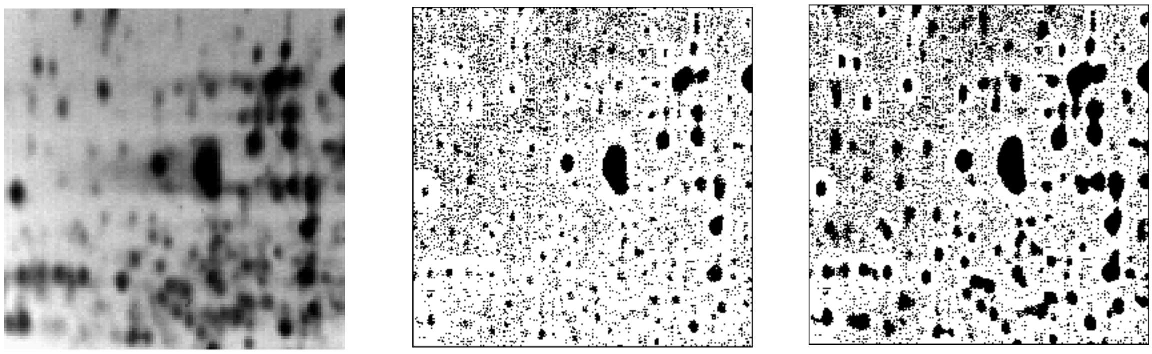
Apply the extraction of extended minima of increasing heights to the ELECTROP image:

```
imdisplay g1 "electrop"
extminima g1 b1 50
imdisplay b1 "depth 50"
extminima g1 b2 100
imdisplay b2 "depth 100"
```

(a) (b) (c)

extended regional minima

(a) original image, (b) extended minima of height 50, (c) of height 100



SUMMARY

This chapter of exercises should have allowed you to introduce the following transformations into the dictionary:

maxima s d
regional maxima of the greytone image s into the binary image d.
minima s d
regional minima of the greytone image s into the binary image d.
extmaxima s d h
extended regional maxima of height h of the greytone image s into the binary image d.
extminima s d h
extended regional minima of height h of the greytone image s into the binary image d.
fgrain s d
the upper left connected component of s is stored in d. This connected component is removed from s.

Chapter 7

SKELETONS

The following exercises introduce morphological transformations which are linked to the concept of maximal ball. This notion is the starting point of the definition of the skeleton of binary sets.

EXERCISES

Exercise n° 1: Ultimate erosion of a (binary) set

Let X be a set. The ultimate erosion of X is defined by:

$$U(X) = \bigcup_n \{x \in X \ominus nH; d_{X \ominus nH}(x, X \ominus (n+1)H) = +\infty\}$$

U(X) is then the set of the connected components resulting from the successive erosions of X that cannot be reconstructed after the erosion of the size immediately larger.

1) Design the algorithm and program the ultimate erosion of a set X.

2) Apply it to the CELLS image. Compute the number of cells in the aggregate.

3) Try to determine the limits of this transformation as a tool for separating particles.

[procedure **binultim**]

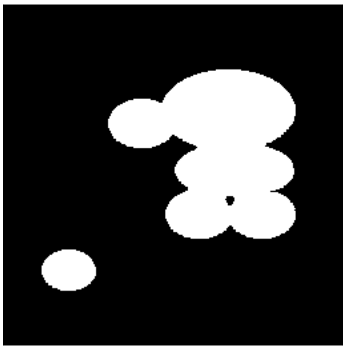
Solution

1) The ultimate erosion is programmed as follows. To determine the points of the successive eroded sets at an infinite geodesic distance from the eroded sets of the size immediately smaller we use the **binbuild** procedure.

```
deproc binultim binultim s d
syntax "binultim binin binout"
 int w i ;
 w := imalloc 1
 imcopy s w
 i := 1
 while i do
  ero w d 1
  build w d
  imdiff w d d
  ero w w 1
  i := imvolume w
  imor d d w
 end
 imfree w
end
```

Note that the ultimate eroded sets detected at each step of this procedure are added to the eroded set. This variation with respect to the definition does not modify the result, but has the advantage of not requiring extra memory.

2) To test the algorithm, proceed to the following steps after the CELLS image has been loaded into b1:

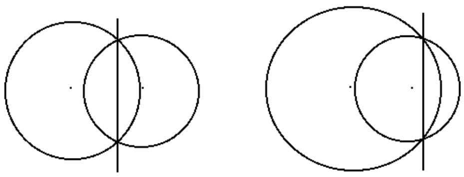


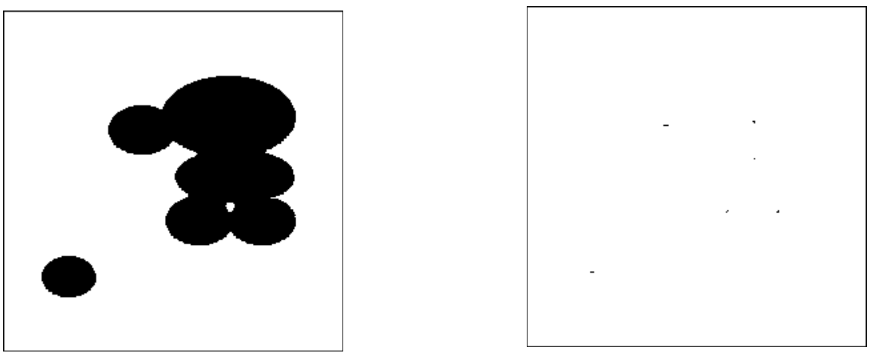
CELLS

binultim b1 b2
print cnumber b2

If all is correct, the printed number should be 6.

3) The separation by ultimate erosion provides satisfactory results only when the aggregate is composed of round particles which do not interpenetrate too much. In the simple case where the aggregate is made of disks, ultimate erosion works if the radical axis of two connected disks separate their centers:





(a) (b)

Ultimate erosion: (a) original, (b) ultimate eroded set

Exercise n° 2: Skeleton by maximal balls (binary case)

Let X be a set. A ball of radius λ included in X is said to be a maximal ball if and only if no ball of a radius strictly larger and containing the ball of radius λ can be found in X.

B_λ maximal: $B_\lambda \subset X$; there exists no B_μ , $\mu > \lambda$, $B_\lambda \subset B_\mu \subset X$

The locus of the centers of the maximal balls of X, S(X) is called the skeleton of X. When X is defined on the hexagonal grid, the notion of maximal ball is replaced by that of maximal hexagon. The purpose of this exercise is to define an algorithm which performs the skeleton S(X) of X.

1) Let $X \ominus nH$, be the erosion of size n of X. Prove that if x is a point of $X \ominus nH$ which does not belong to the open set $(X \ominus nH)_H$, this point is the center of a maximal hexagon of size n.

2) Derive the algorithm performing the skeleton S(X) of X.

3) To any point x of S(X), can be associated the radius r(x) of the maximal hexagon centered in x. The function r(x) of support S(X) is called a quench function. Verify that the pair (S(X),r(x)) suffices to reconstruct the set X:

$$X = \bigcup_{x \in S(X)} H(x, r(x))$$

4) Compare the skeleton of the set X with its ultimate erosion.

5) What are the drawbacks of this transformation in the digital case?

[procedure **binopenskel**]

Solution

1) Let x be a point such that:

$$x \in (X \ominus nH) \text{ et } x \notin (X \ominus nH)_H$$

If x belongs to the eroded set of size n, x is by definition the center of a hexagon of size n included in X.

Assume that $X \in (X \ominus nH)_H$. x is then included in a hexagon of size 1 included in the eroded set $(X \ominus nH)$.

A size n dilation of this hexagon results in a hexagon of size n+1 included in X. Hence the hexagon of size n centered in x is covered by the hexagon of size n+1. Therefore it cannot be maximal. Conversely, let us assume that x, though not belonging to the open set$(X \ominus nH)_H$, is not the center of a maximal hexagon of size n. Then, it is included in a hexagon of size m>n covering the hexagon of size n centered in x. The erosion of this hexagon of size m by a hexagon of size n gives a hexagon of size $m - n \geq 1$ which contains x. Then, $x \in (X \ominus nH)_H$, which is contradictory.

Therefore, it can be said that:

a necessary and sufficient condition for a point x to be the center of a maximal hexagon of size n is that it belongs to the set:

$$(X \ominus nH)/(X \ominus nH)_H$$

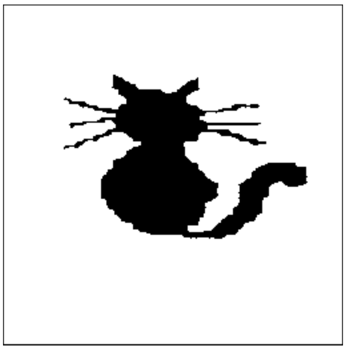
2) According to what has just been stated, in order to obtain the skeleton S(X), it is sufficient to perform the preceding set difference for all the possible sizes of erosions, i.e.:

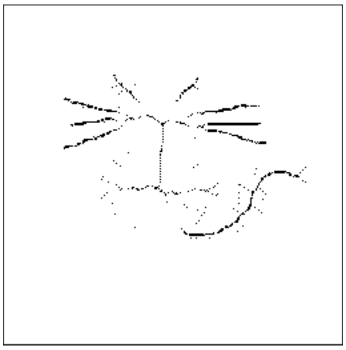
$$S(X) = \bigcup_{n=0}^{\infty} [(X \ominus nH)/(X \ominus nH)_H]$$

Here is an example of a procedure which realizes this type of skeleton:

```
deproc binopenskel binopenskel s d
syntax "binopenskel binin binout"
 int w i ;
 w := imalloc 1
 i := 1
 imcopy s w
 while i do
  open w d 1
```

```
      imdiff w d d
      ero w w 1
      i := imvolume w
      imor d w
    end
    imfree w
  end
```





(a) (b)

Non connected skeleton
(a) original image, (b) skeleton by openings

3) Prove that any point x of X belongs to the maximal ball.

Indeed, if this were not the case, then there would exist a ball centered in x and contained in X (the ball may be reduced to x itself). This ball is not included in any other ball included in X. (Otherwise this including ball would be itself either maximal, or included in a maximal ball). Therefore the ball centered in x is the maximal ball. x belongs to a maximal ball of X, which contradicts the hypothesis.

$$\forall x \in X, x \in \text{maximal ball of } X$$

The union of all the maximal balls of X is then equal to X.

4) It can be easily verified that the ultimate erosion of a set is contained in the skeleton.

Indeed, let x be a point of the eroded set X. Assume this point has appeared in the erosion of size n. $x \in (X \ominus nH)$. This point x is at an infinite geodesic distance from $X \ominus (n+1)H$. In other words, the dilated set $[X \ominus (n+1)H] \oplus H$ does not encounter x. (since this dilated set represents the set of the points of the space which are at a geodesic distance smaller than or equal to 1 from $X \ominus (n+1)H$).

But:

$$[X \ominus (n+1)H] \oplus H = (X \ominus nH)_H$$

Then:

$$x \in (X \ominus nH)/(X \ominus nH)_H \Rightarrow x \in S(X)$$

Exercise n° 3: Distance function and maxima

This exercise consider again the notion of distance function introduced in chapter 2 and the corresponding procedure **distance**.

1) Prove that the local maxima of the distance function are the points of the skeleton by maximal balls. Observe it on the COFFEE image.

2) Prove that the regional maxima of the distance function are the ultimate eroded sets. Observe it on the COFFEE image. What can be the interest of the extended regional maxima of the distance function? Test it on the COFFEE image.

Solution

1) Let us prove the proposition.

Let x be a point of the skeleton by hexagonal opening. x is the center of a maximal hexagon of size n. It results that dist(x)=n+1.
Assume there exists a neighbor y of x such that dist(y) > dist(x).
Then H^{n+1} (hexagon of center y of size n+1) $\subset$ X.
But $H_x^n \subset H_y^{n+1}$. Then there exists a hexagon of size n+1 that contains x and that is included in X, which contradicts the hypothesis according to which H is a maximal hexagon.
Then $\forall$y neighbor of x, dist(y) $\leq$ dist(x). x is then a local maximum of the distance function.

Conversely, let x be a local maximum of the hexagonal distance function.
Let n = dist(x)-1. H_x^n is the largest hexagon of center x included in X.
Assume that H^n is not maximal, i.e.:

$$\exists H_y^{n'} \text{ tel que } H_x^n \subset H_y^{n'} \text{ et } H_x^n \neq H_y^n$$

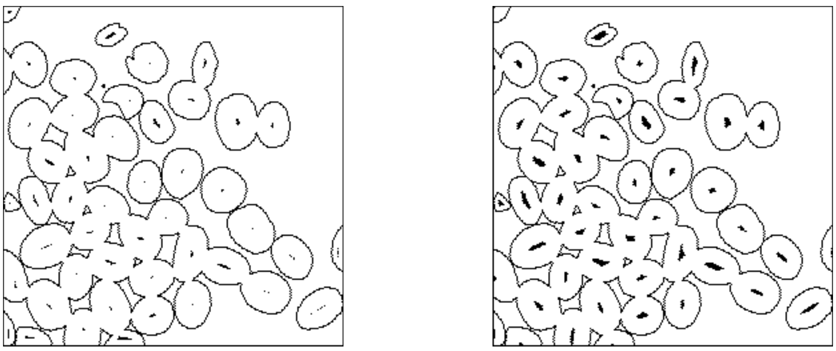
$H_y^{n'} \ominus H_x^n \neq \emptyset$ and it is a hexagon H_z^m. Moreover $m \geq 1$ for $H_y^{n'} \neq H_x^n$ et $x \in H_z^m$.

The erosion of H_z^m by a elementary hexagon is a hexagon H_z^{m-1} and $\forall y \in H_z^m / H_z^{m-1}$, y there exists a neighbor in H_z^{m-1}. However $x \notin H_z^m \ominus H$(otherwise $H_x^{n+1} \subset X$, but H_x^n is the largest hexagon of center x included in X) and $H_z^m \ominus H \neq \emptyset$. then x has a neighbor y in H_z^{m-1}. It results that $H_y^{n+1} \subset X$ and hence $\text{dist}(y) \geq n+2 = \text{dist}(x)+1$. This contradicts the assumption that x is a local maximum of the distance function. Then H_x^n is maximum. Q.E.D.

Let us verify this after the COFFEE image has been loaded into b1:

binopenskel b1 b2
distance b1 g1
dil g1 g2 1
imsub g2 g1 g2
imthresh g2 0 0 b3

2) An ultimate eroded set is a connected component of the eroded set of size n that cannot be restored from the eroded set of size n+1. However the eroded set of size n is the threshold between n+1 and 255 of the distance function. Then, the ultimate eroded sets are but the regional maxima of the distance function



(a) (b)

(a) ultimate erosion, (b) extended regional maxima of height 2

The COFFEE image shows that some grains are marked by several ultimate erosions, which is undesirable for the following exercises (segmentation). The extended regional maxima can help us to connect these markers, so that we obtain a better segmentation.

Let us verify on the COFFEE image the matching between the ultimate eroded sets and the regional maxima of the distance function. The COFFEE image is previously loaded into b1.

distance b1 g1
maxima g1 b2
binultim b1 b3

SUMMARY

This chapter of exercises should have allowed you to introduce the following transformations into the dictionary:

binultim s d
ultimate erosion of the binary image s into the binary image d.
binopenskel s d
skeleton by maximal balls of the binary image s into the binary image d.

Chapter 8

THINNINGS AND THICKENINGS

8.1. Introduction

The transformations developed in the following exercises are more sophisticated. According to our previous comparison, they are the true "machine-tools" of MM. They are at the meeting point of geodesy and homotopy. We have already handled geodesic transformations, therefore we shall only recall the notion of homotopy, and especially that of homotopic transformation.

8.2. Binary thinnings, reminder

Let $T = (T_1, T_2)$ be a two-phase structuring element. The hit-or-miss transformation of a set X by T is equal to:

$$X * T = (X \ominus \check{T}_1) \cap (X^c \ominus \check{T}_2)$$

The thickening of X by T is equal to:

$$X \odot T = X \cup (X * T)$$

and the thinning is defined by:

$$X \bigcirc T = X/(X * T)$$

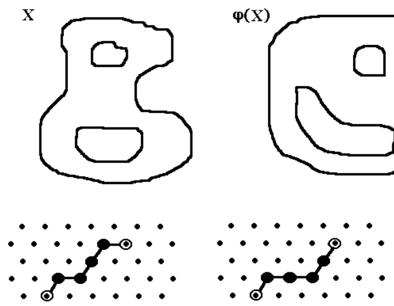
Thinning and thickening are both dual transformations:

$$(X^c \odot T)^c = [X^c \cup (X^c * T)]^c = X \cap (X^c * T)^c = X / (X * T') = X \bigcirc T'$$

where $T' = (T_2, T_1)$.

8.3. Homotopy

Two paths of a set X are homotopic if it is possible to superpose one another by a sequence of continuous deformations. "Continuous" means without cut and that all intermediary paths are included in X.



Homotopic paths on the hexagonal grid

By extension, a transformation Ψ which preserves homotopy is said to be homotopic. Intuitively, a homotopic transformation Y = Ψ(X) transforms the set X in a set Y that can be superposed on X by a continuous deformation.

When the set X is digitized according to a (square or hexagonal) grid, matching paths is easy since any path can be defined as the concatenation of elementary edges. A homotopic transformation does not break any paths.

8.4. Greytone thinnings

Let $m(x) = \sup_{y \in T_{2x}} f(y)$ and $M(x) = \inf_{y \in T_{1x}} f(y)$.

The thickening of f by T = (T_1 ,T_2) is defined by :

$$(f \odot T)(x) = M(x), \text{ iff } m(x) \leq f(x) < M(x)$$
$$(f \odot T)(x) = f(x), \text{ if not.}$$

The thinning of f by T = (T_1 ,T_2) is defined by:

$$(f \bigcirc T)(x) = m(x), \text{ iff } m(x) < f(x) \leq M(x)$$
$$(f \bigcirc T)(x) = f(x), \text{ if not.}$$

EXERCISES

Exercise n° 1

On the hexagonal grid, the most interesting structuring elements, T = (T_1,T_2) are those defined on the elementary hexagon:

$$T_1 \subset H, T_{2 \subset H}$$

We can even write: $T_1 \cap T_2 = \emptyset$. Indeed, T_1 and T_2 must have no common point if we want X * T to be different from an empty set.

1) Program the binary and greytone thickening and thinning by any structuring element defined on the elementary hexagon. The hit-or-miss transformation **imhitormiss** is already available.

2) Apply the algorithms to binary images. In particular, perform the following operations :

- delete the white and black isolated points in the NOISE image.
- contour a set in only one transformation.

Solution

1) binthin and binthick procedures

imhitormiss is already present in the dictionary. To program thinning:

```
deproc binthin binthin se1 se0 s d
syntax "binthin se_for_1 se_for_0 binin binout"
  int w oldedge;
  w := imalloc 1
  oldedge := edge
   imsetedge 0
   imhitormiss s se1 se0  w
  imdiff s w d
```

```
  imsetedge oldedge
  imfree w
end
```

The procedure can be iterated for all the rotations of the structuring element:

```
  deproc thinturn thinturn se1 se0 s d
  syntax "thinturn se_for_1 se_for_0 binin binout"
    imcopy s d
    for 1 to ngbnb do
      binthin se1 se0 d d
      se0 := serotate se0
      se1 := serotate se1
    end
  end
```

next, thickening :

```
deproc binthick binthick se1 se0 s d
syntax "binthick se_for_1 se_for_0 binin binout"
  int w oldedge ;
  w := imalloc 1
  oldedge := edge
  imsetedge 0
  imhitormiss s se1 se0  w
  imsup s w d
  imsetedge oldedge
  imfree w
end
```

Then, the procedure with rotations:

```
  deproc thickturn thickturn se1 se0 s d
  syntax "thickturn se_for_1 se_for_0 binin binout"
    imcopy s d
    for 1 to ngbnb do
      binthick se1 se0 d d
      se0 := serotate se0
      se1 := serotate se1
    end
  end
```

In order to program equivalent greytone procedures, first define the dilation and erosion of a greytone image by any structuring element se (defined on a unitary hexagon).

```
  deproc greyseero greyseero se s d
 syntax "greyseero struct_elt greyin greyout"
   int i j w;
```

```
      w := imalloc imdepth s
      imset impixmax w w
      j := 0
      i := 1
      for 0 to ngbnb do
        if (se && i) then
          iminfngb s w j 1
        end
        i := (i * 2)
        ++ j
     end
     imcopy w d
     imfree w
  end

   deproc greysedil greysedil se s d
   syntax "greysedil struct_elt greyin greyout"
      int i j w;
      w := imalloc imdepth s
      imset 0 w
     j := 0
     i := 1
     for 0 to ngbnb do
       if (se && i) then
          imsupngb s w j 1
       end
       i := (i * 2)
      ++ j
    end
    imcopy w d
    imfree w
   end
```

Then thinning and thickening are written this way:

```
     deproc greythin greythin sei ses s d
     syntax "greythin se_for_inf se_for_sup greyin greyout"
        int m M bw1 bw2 gw;
       m := imalloc imdepth s
       M := imalloc imdepth s
      gw := imalloc imdepth s
      bw1 := imalloc 1
      bw2 := imalloc 1
      greysedil ses s m
      greyseero sei s M
      imsub s M M
      imthresh M 1 impixmax s bw1           { bw1 = where s > M }
      iminv bw1 bw1                         { bw1 = where s <= M }
```

```
    imsub s m M                              { M is no longer used }
    imthresh M 1 impixmax s bw2             { bw2 = where m < s }
    imsup bw2 bw1 bw1                 { bw1 = where m < s <= M }
    immask bw1 0 impixmax s gw
    iminf gw m m
     imcopy s d
    iminv gw gw
    iminf gw d d
    imsup m d d
    imfree m
    imfree M
   imfree gw
   imfree bw1
   imfree bw2
 end

 deproc greythick greythick sei ses s d
 syntax "greythick se_for_inf se_for_sup greyin greyout"
    int m M bw1 bw2 gw;
    m := imalloc imdepth s
    M := imalloc imdepth s
    gw := imalloc imdepth s
    bw1 := imalloc 1
    bw2 := imalloc 1
    greysedil ses s m
    greyseero sei s M
    imsub m s m
    imthresh m 1 impixmax s bw1{ bw1 = where m > s }
    iminv bw1 bw1                { bw1 = where m <= s }
    imsub M s m                  { m is no longer used }
    imthresh m 1 impixmax s bw2{ bw2 = where s < m }
    iminf bw2 bw1 bw2            { bw1 = where m <= s < M }
    immask bw1 0 impixmax s gw
    iminf gw M M
    imcopy s d
    iminv gw gw
   iminf gw d d
   imsup M d d
   imfree m
   imfree M
   imfree gw
   imfree bw1
   imfree bw2
 end
```

2) Examples

- Deletion of isolated points

Load the NOISE image into dans b1 and proceed as follows:

```
binthin 1 126 b1 b1
binthick 126 1 b1 b1
```

Now, display the result.

- Set contour

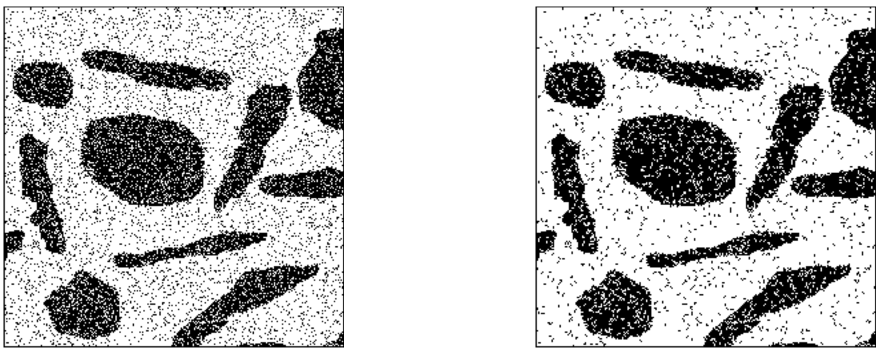
Given:

$$C(X) = X/(X \ominus H) = X \cap (X \ominus H)^c$$

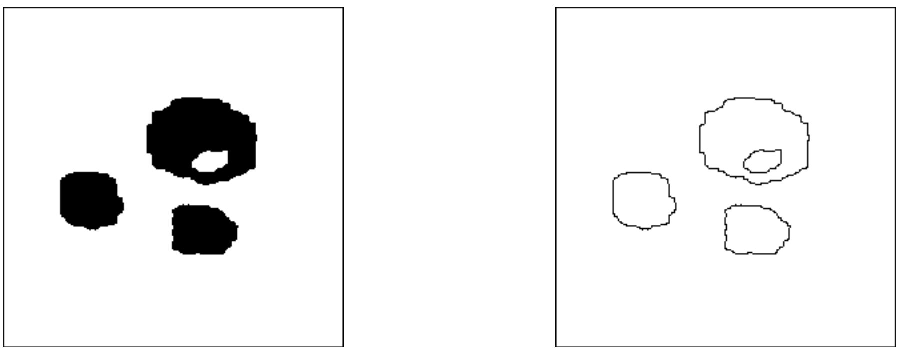
Then:

$$C(X) = X \odot T \text{ , avec } T = (H,\varnothing)$$

```
deproc bincontour bincontour s d
syntax "bincontour binin binout"
  binthin 127 0 s d
end
```



Deletion of isolated points



Contour of a set

Exercise n° 2: Geodesic thickenings and thinnings

Let X be a set included in Z. The geodesic thickening of X by a structuring element T included in the elementary hexagon can be defined by:

$$(X \odot T) = (X \odot T) \cap Z$$

1) Program this transformation.

2) Since thinning is the dual operation of thickening, geodesic thinning is defined by:

$$(X \bigcirc T) = Z/[(Z/X) \odot T']$$

Simplify and program this transformation.

Solution

1) Geodesic thickening

```
deproc bingdsthick bingdsthick se1 se0 s m d
syntax "bingdsthick se_for_1 se_for_0 binin binmask binout"
  binthick se1 se0 s d
  imand m d
end

deproc gdsthickturn gdsthickturn se1 se0 s m d
syntax "gdsthickturn se_for_1 se_for_0 binin binmask binout"
   int w ;
   w  := imalloc 1
   imcopy s w
   for 1 to ngbnb do
      bingdsthick se1 se0 w m w
      se0 := serotate se0
      se1 := serotate se1
   end
  imcopy w d
  imfree w
end
```

2) Geodesic thinning

Geodesic thinning is achieved by using a thickening by the dual structuring element, so as to avoid the problems due to border effects already mentioned about geodesic erosions.

```
deproc bingdsthin bingdsthin se1 se0 s m d
syntax "bingdsthin se_for_1 se_for_0 binin binmask binout"
  bindiff m s d
  binthick se0 se1 d d
  bindiff m d d
end

 deproc gdsthinturn gdsthinturn se1 se0 s m d
 syntax "gdsthinturn se_for_1 se_for_0 binin  binmask binout"
    int w ;
    w := imalloc 1
    imdiff m s w
    gdsthickturn se0 se1 w m w
    imdiff m w d
    imfree w
 end
```

Exercise n° 3

1) Find all possible configurations (up to a rotation) for the neighboring of a point in the hexagonal grid (they are 14).

2) Prove that the relation of homotopy for two paths C_1 and C_2 with same origin and same extremity included in a set X is a relation of equivalence.

3) From this, deduce which of these configurations generate a homotopic thinning.

Solution

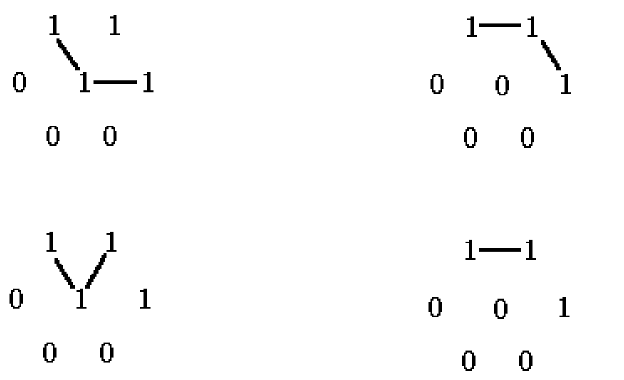
1) Below are displayed the 14 configurations (the central point is assumed to be 1):

```
  0   0       1   0       1   1       1   0
0   1   0   0   1   0   0   1   0   0   1   1
  0   0       0   0       0   0       0   0
   (1)         (2)         (3)         (4)

  1   0       1   0       1   1       1   1
0   1   0   0   1   0   0   1   1   0   1   0
  0   1       1   0       0   0       0   1
   (5)         (6)         (7)         (8)

  1   1       1   0       1   1       1   1
0   1   0   0   1   1   0   1   1   0   1   1
  1   0       1   0       0   1       1   0
   (9)         (10)        (11)        (12)

              1   1       1   1
            0   1   1   1   1   1
              1   1       1   1
               (13)        (14)
```

2) Obvious. The homotopy relation is an equivalence relation.

3) The thinning by one of the 14 possible configurations replaces the central point 1 by 0. To extract the configurations which generate homotopic thinnings, it suffices to verify that any path which passes through the central point can be replaced, after thinning, by an equivalent path in the hexagonal neighborhood.

This can be illustrated in the case of configuration n° 6. It suffices to verify that all the paths are preserved:



Initial paths　　　Equivalent paths

Configuration 6

Using the same procedure for the 13 other configurations, we can see that only those which are formed by one connected component generate homotopic thinning, namely: configurations 2, 3, 6, 10 and 13.

Exercise n° 4

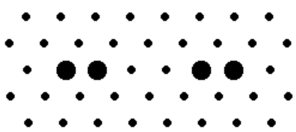
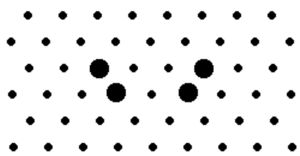
1) Program the connected skeleton by using the structuring elements L, M and D. Compare the results.

2) Indicate the transformations allowing to extract the characteristic points of a skeleton:
- extremities
- n-uple points
- single poins

L	M	D
1 1	1 1	1 •
• • •	1 • •	• • 0
0 0	• 0	0 0

3) The direction of rotation and the starting orientation of the structuring elements used for the different skeletons are totally arbitrary. Consequently, more or less important variations occur in the resulting skeleton. These variations may even generate artifacts.

Generate one of the two following images:

Apply to this image a skeleton of type L by thickening, and judge the result. Is it possible to improve the algorithm?
[procedures **Lthin** ; **Mthin** ; **Dthin** ; **multpoint** ; **endpoint**]

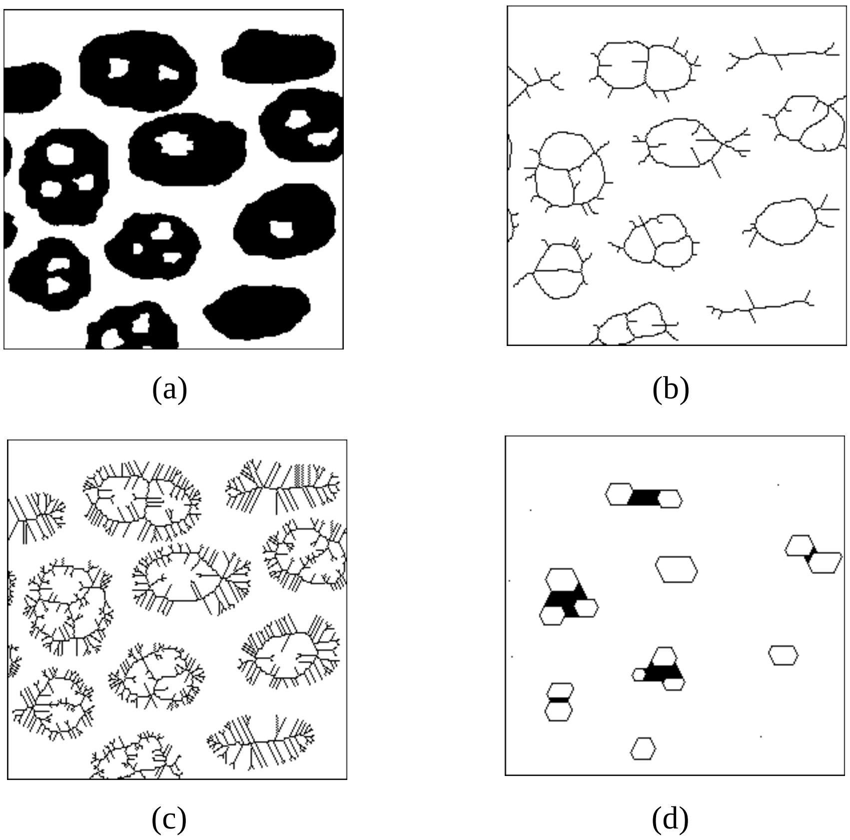
Solution

1) The procedure used to perform the skeleton by means of element L.

This skeleton resorts to the rotation of the structuring elements L and can be defined in two steps. First, a thinning procedure by the rotation of a structuring element se is defined. Next a general algorithm for the skeleton is defined by means of this procedure.

```
deproc thin thin se1 se0 s d
syntax "thin se_for_1 se_for_0 binin binout"
   int i1 i2 ;
   imcopy s d
   i1 := imvolume d
   if (edge = 0) then
     while (i1 <> i2) do
        i2 := i1
        thinturn se1 se0 d d
        i1 := imvolume d
```

```
        end
      else
      iminv d d
      while (i1 <> i2) do
         i2 := i1
         thickturn se0 se1 d d
         i1 := imvolume d
      end
    iminv d d
    end
  end
```



Skeletons, (a) original, (b) skeleton L, (c) skeleton M, (d) skeleton D

The skeleton L is written thus:

```
deproc Lthin Lthin s d
syntax "Lthin binin binout"
   if (grid <> 0) then
     thin 66 24 s d
   else
     thin 262 112 s d
   end
end
```

The procedures **Mthin** and **Dthin** are defined in the same way: it suffices to change the structuring elements.

```
deproc Mthin Mthin s d
syntax "Mthin binin binout"
  if (grid <> 0) then
     thin 56 2 s d
  else
     thin 120 258 s d
  end
end

deproc Dthin Dthin s d
syntax "Dthin binin binout"
  if (grid <> 0) then
     thin 2 56 s d
  else
     thin 258 120 s d
  end
end
```

Note that the three homotopic transformations give quite different results :
- **Mthin** produces a rather high number of dendrites ;
- **Dthin** reduces any simply connected set to one point ;
- **Lthin** is the only transformation whose behavior corresponds to the intuitive idea one may have of the skeleton of a set.

2) Extracting single points of the skeleton

- <u>Extremities</u>

The extremities of the skeleton correspond to the configurations (see exercise n° 3) 1, 2, 3 (up to a rotation).

In order to simplify the structuring element, note that:

```
  0   0         1   0         1   1         •   •
0   1   0  ∪  0   1   0  ∪  0   1   0  =  0   1   0
  0   0         0   0         0   0         0   0
```

```
deproc endpoints endpoints s d
syntax "endpoint binin binout"
  int se0 se1 w w1 oldedge ;
  w := imalloc 1
  w1 := imalloc 1
  oldedge := edge
  imsetedge 0
  if (grid <> 0) then
    se0 := 30
    se1 := 1
  else
    se0 := 62
    se1 := 1
```

```
    end
    for 1 to ngbnb do
      imhitormiss s se1 se0 w
      imor w w1 w1
      se0 := serotate se0
      se1 := serotate se1
   end
   imcopy w1 d
   imsetedge oldedge
   imfree w1
   imfree w
end
```

- Multiple points

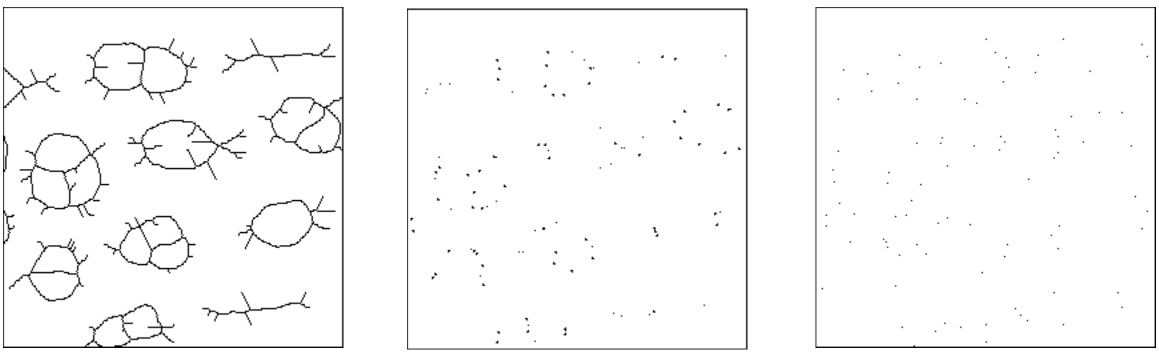
Only the configurations 6, 7, 8, 9, 10, 11, 12, 13 and 14 correspond to multiple points. Since the extremities correspond to configurations 1, 2 and 3, the multiple points can be obtained in the following way:

a) Thinning by rotating 4 and 5 configurations.

b) Suppression of extremities.

```
  deproc mulpoints mulpoints s d
 syntax "mulpoints binin binout"
    int w w1 se0 se1 oldedge ;
    w  := imalloc 1
    w1 := imalloc 1
   oldedge := edge
   imsetedge 0
   if (grid = 1) then
     endpoints s w1
     se1 := 11
     se0 := 116
     for 1 to 6 do
       imhitormiss s se1 se0 w
       imsup w w1 w1
       se0 := serotate se0
       se1 := serotate se1
    end
    se1 := 19
    se0 := 108
   for 1 to 3 do
     imhitormiss s se1 se0 w
     imsup w w1 w1
     se0 := serotate se0
     se1 := serotate se1
  end
  imdiff s w1 w1
 else
  imset impixmin w1 w1
```

```
  se1 := 169
  se0 := 340
  for 1 to 4 do
    imhitormiss s se1 se0 w
    imsup w w1 w1
    se0 := s4rotate se0
    se1 := s4rotate se1
  end
  se1 := 169
  se0 := 2
  for 1 to 4 do
    imhitormiss s se1 se0 w
    imsup w w1 w1
    se0 := s4rotate se0
    se1 := s4rotate se1
   end
   se1 := 170
   se0 := 1
   for 1 to 4 do
     imhitormiss s se1 se0 w
     imsup w w1 w1
     se0 := s4rotate se0
     se1 := s4rotate se1
   end
  end
  imcopy w1 d
  imsetedge oldedge
  imfree w
  imfree w1
end
```



(a) (b) (c)

(a) original, (b) multiple points, (c) extremities

- Isolated points

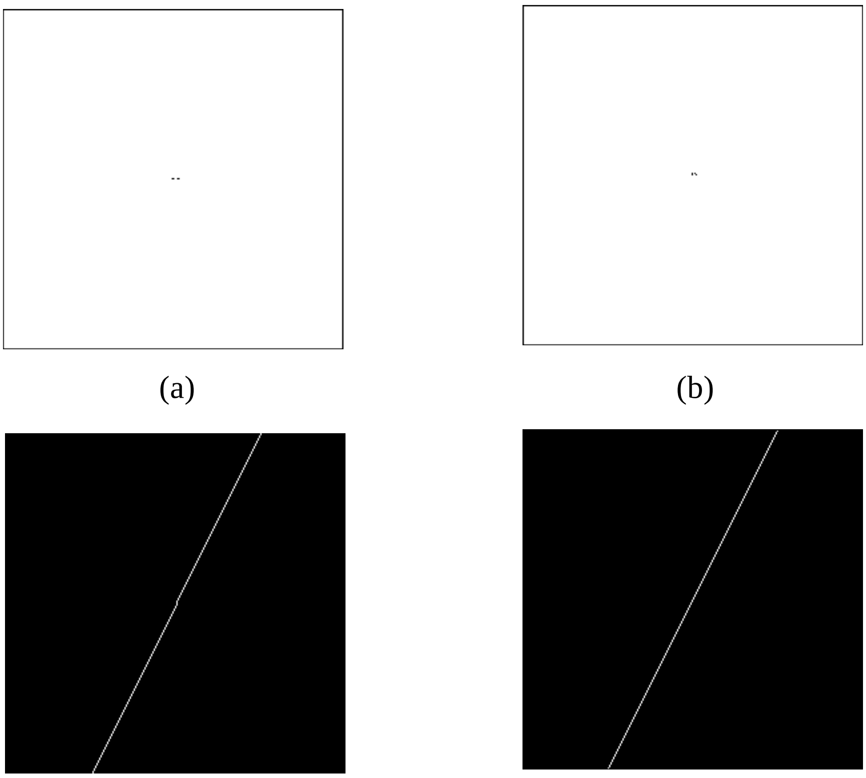
Obvious (it is configuration 1).

3) To generate test images into image b1 (for example), do:

imset 0 b1
imwritepix 1 127 127 b1
imwritepix 1 128 127 b1
imwritepix 1 131 127 b1
imwritepix 1 132 127 b1

and for the second:

imset 0 b1
imwritepix 1 127 127 b1
imwritepix 1 130 127 b1
imwritepix 1 127 126 b1
imwritepix 1 129 126 b1



(a) (b) (c) (d)
Behavior of the skeleton
(a), (b) Tested configurations, (c), (d) Corresponding skeletons

The result of the skeleton by thickening (write a procedure **Lthick** similar to **Lthin**) is given above:

This divergence comes from the fact that the skeleton algorithm is not isotropic, since we work one direction after the other.

More efficient algorithms exist. They use in particular structuring elements defined on hexagons of size 2, which enables them to consider all rotations simultaneously.

deproc thick thick se1 se0 s d

```
syntax "thick se_for_1 se_for_0 binin binout"
  int i1 i2 ;
   imcopy s d
  i1 := imvolume d
  if (edge = 1) then
    while (i1 <> i2) do
      i2 := i1
      thickturn se1 se0 d d
      i1 := imvolume d
    end
  else
    iminv d d
   while (i1 <> i2) do
     i2 := i1
     thinturn se0 se1 d d
     i1 := imvolume d
    end
    iminv d d
  end
end

deproc Lthick Lthick s d
 syntax "Lthick binin binout"
   if (grid <> 0) then
      thick 24 66 s d
   else
      thick 112 262 s d
   end
 end

 deproc Mthick Mthick s d
 syntax "Mthick binin binout "
   if (grid <> 0) then
     thick 56 2 s d
   else
     thick 120 258 s d
   end
 end

 deproc Dthick Dthick s d
 syntax "Dthick binin binout"
    if (grid <> 0) then
       thick 2 56 s d
    else
       thick 258 120 s d
    end
 end
```

Exercise n° 5

The exercise 3 has allowed you to define homotopic thinnings on an elementary hexagon.

1) Among the selected configurations, Study their effect on the length of the homotopic paths, after thinning. Indicate in particular why the following configuration is special.

$$\begin{array}{ccccc} & \bullet & & \bullet & \\ 0 & & \bullet & & \bullet \\ & 0 & & 0 & \end{array}$$

2) Deduce from this an algorithm for detecting the geodesic center of simply connected sets. Program this algorithm.
[procedures **Dthin** ; **gdsthin** (applied to D)]

Solution

1) We know that with each point of a set X that is simply connected (without holes), we can associate the value of the greatest geodesic distance between this point and any other point of X. Thus, we define a function on X whose maxima correspond to the extremities of X and the minimum (unique) to the geodesic center. Consider the following configuration:

$$\begin{array}{ccccc} & 1 & & 1 & \\ 0 & & 1 & & 1 \\ & 0 & & 0 & \end{array}$$

that belongs to X and let n be the value of the function previously defined at the central point. It can be shown that in this configuration there always exists a point of the contour for which the function value is n as well:

$$\begin{array}{ccccc} & n & & n' & \\ 0 & & n & & n'' \\ & 0 & & 0 & \end{array} \quad \text{(for instance)}$$

We admit this result (which can be proved by studying the different possible geodesic paths starting from the central point).

Its immediate consequence is that the thinning of X direction after direction, using this structuring element produces a set Y whose greatest distance function, though reduced of certain points, is identical on Y to the one defined on X.

Hence the two sets have the same extremities (except for the points that have been possibly eliminated) and the same geodesic center.

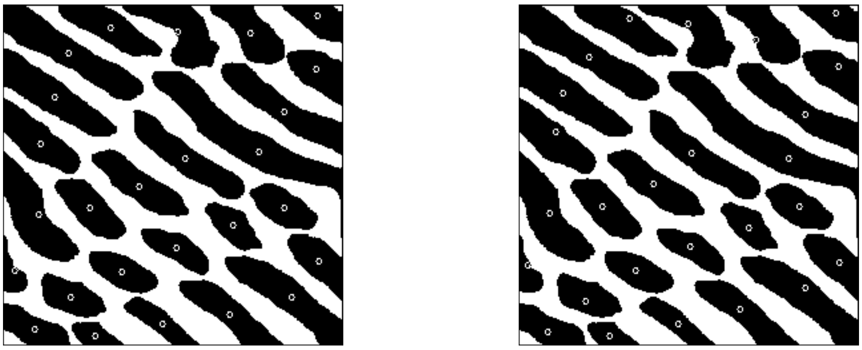
2) Trick question! Indeed one could think that the thinning algorithm using the structuring element D in which the three configurations:

$$\text{(a)}\ \begin{array}{ccccc} & 1 & & 1 & \\ 0 & & 1 & & 1 \\ & 0 & & 0 & \end{array} \qquad \text{(b)}\ \begin{array}{ccccc} & 1 & & 1 & \\ 0 & & 1 & & 0 \\ & 0 & & 0 & \end{array} \qquad \text{(c)}\ \begin{array}{ccccc} & 1 & & 0 & \\ 0 & & 1 & & 0 \\ & 0 & & 0 & \end{array}$$

that compose it have been separated, would be more suitable to produce the geodesic center. Indeed, element (a) does not modify the greatest distance function, whereas (c) systematically reduces it by 1. Unfortunately, the behavior of (b) is sometimes similar to that of (a), and sometimes of (c).

However the placing of the centroid obtained by **Dthin** with respect to the actual geodesic center can be improved by applying the procedure below.

Note there exists an exact algorithm to obtain the geodesic center. It uses a special type of morphological transforms called propagations.



(a) (b)

Geodesic centers
(a) with skeleton D, (b) with the described procedure

```
deproc gdscentre gdscentre s d
syntax "gdscentre binin binout"
   int se10 se11 se20 se21 se30 se31 w w1 ;
   if (grid <> 1) then makeerror 10001 end
     w := imalloc 1
     w1 := imalloc 1
     se10 := 112
     se11 := 14
     se20 := 120
     se21 := 6
     se30 := 124
     se31 := 2
     imcopy s d
     imset 0 w
     while (imcompare w d w <> -1) do
        while (imcompare w d w <> -1) do
           imcopy d w
           thinturn se11 se10 d d
        end
       thinturn se21 se20 d d
       thinturn se31 se30 d d
     end
    imfree w
    imfree w1
  end
```

Exercise n° 6

Program, by means of geodesic transformations, the geodesic skeleton by thickening. Verify in particular that only the structuring element:

allows to obtain a correct transformation.

Solution

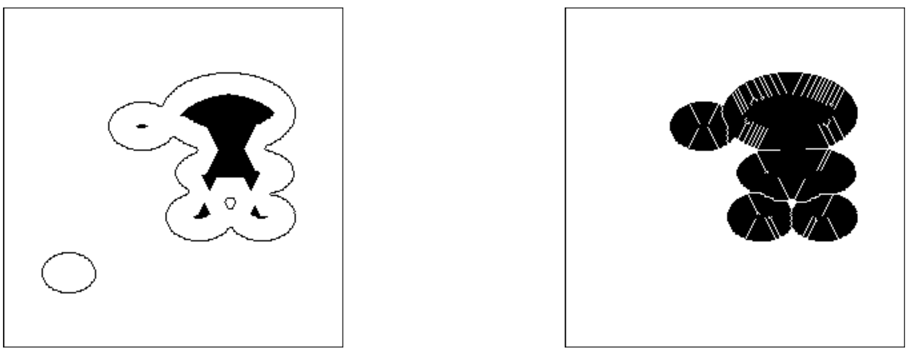
We have to use the structuring element M. Indeed, it is the only one able to penetrate the areas without thickness of the geodesic set X:

	0	0
•	•	0
	1	•

With L, the result would be:

This skeleton will be defined in two steps as for the euclidean skeleton:

```
deproc gdsthick gdsthick se1 se0 s m d
syntax "gdsthick se_for_1 se_for_0 binin binmask binout"
  int i1 i2 w ;
  w := imalloc 1
  imcopy s w
  i1 := imvolume w
  while (i1 <> i2) do
     i2 := i1
     gdsthickturn se1 se0 w m w
     i1 := imvolume w
  end
  imcopy w d
  imfree w
end
```



(a) (b)

Geodesic skeleton, (a) set to be skeletonized and geodesic set, (b) result

```
deproc gdsMthick gdsMthick s m d
  syntax "gdsMthick binin binmask binout"
     int se0 se1 ;
     se0 := 56
     se1 := 2
     gdsthick se1 se0 s m d
  end
```

Exercice n° 7

The morphological gradient of a function f of $\mathbb{R}^2$ that takes its values on $\mathbb{R}$ in the direction α is defined by:

$$g^\alpha(f) = \lim_{\lambda \to 0} \frac{(f \oplus \lambda L^\alpha) - (f \ominus \lambda L^\alpha)}{2\lambda}$$

where λL^α is a segment of length λ in the direction α. In the digital version, the gradient is defined by:

$$g^\alpha(f) = (f \oplus L^\alpha) - (f \ominus L^\alpha)$$

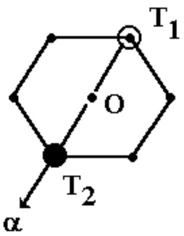
where L^α is the elementary segment in the direction α of the grid.

In the horizontal plane, the gradient azimuth is the direction of the vector $\overrightarrow{grad}(f)$. It can be defined in the directions of the digitization grid.

A new gradient can be defined in the direction α by means of thickenings and thinnings. The directional gradient in the direction α is defined by:

$$g_\alpha(f) = (f \odot T_\alpha) - (f \bigcirc T_\alpha)$$

where T_1 and T_2 form the two-phase structuring element $T_\alpha = (T_1, T_2)_\alpha$.



The maximal directional gradient may occur in several directions simultaneously. Then, we have to compute the most probable direction. To do so, we take into account that computing these gradients implies that three directions at most can be extracted in the hexagonal grid resp. four in the square grid.

1) Find all possible configurations of maximal directional gradients (up to a rotation, and in the hexagonal grid). Their number is 5.

2) In case of non adjacent directions, the gradient is considered to be 0. In case of adjacent directions, a unique direction is selected, which is the mean direction of all present directions. Which configurations do we obtain (up to a rotation)? Their number is 3. What is the resulting effect and how can it be avoided?

3) Program the directional gradient and apply the azimuth to the PETROLE image. [procedure **gradvect**]

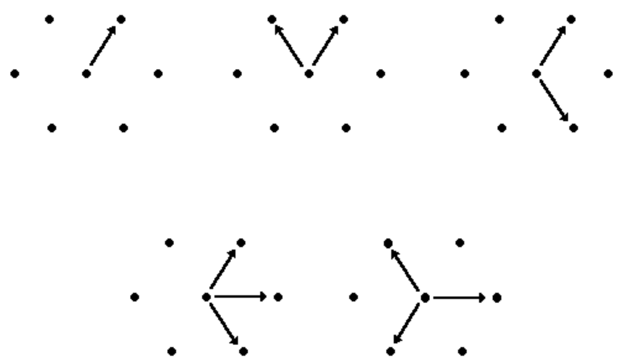
Solution

1) Computing the gradient azimuth requires to compute the directional gradients $g^\alpha(f)$ for all the directions α that can be exhibited on the digitization grid. Then, the gradient

azimuth is the direction α that corresponds to the highest directional gradient. Since the processing is made in hexagonal grid, there are six possible directions for the azimuth. However, several directions of directional gradients may happen to be maximal values. This phenomenon is embarrassing as the gradient vector is unique at any point of the image. It is therefore necessary to correct the rough image of the azimuths obtained by simply detecting the direction (s) of highest directional gradient. In order to do so, a first transformation allows to detect all the directions for which the directional gradient is maximum.

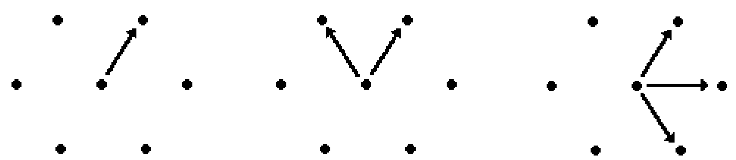
The procedure for computing the directional gradient is given below.

```
deproc dirgradient dirgradient dir s d
 syntax "dirgradient dir greyin greyout"
    int w se1 se2 ;
    w := imalloc imdepth s
    se1 := (2 power dir)
    se2 := setranspose se1
    greythickstep se1 se2 s w
    greythinstep se1 se2 s d
    imsub w d d
   imfree w
 end
```

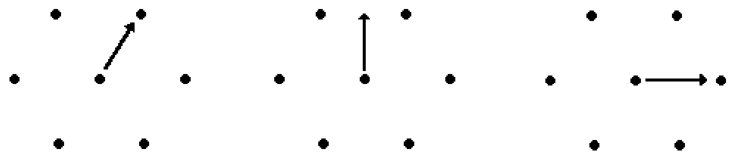


Set of the possible configurations of maximal directional gradients (up to the rotations)

2) Computing these gradients by thickening/thinning implies that three directions at most can be extracted. Next, the various configurations are sorted out. If the marked directions are not adjacent, then the gradient is considered to be zero. Otherwise a unique direction is chosen, which is the mean directions of all present directions. In fact, this case may be summed up into the three situations shown below (up to a rotation):



Initial directions



Selected directions

The second configuration is quite interesting, as in this case the mean direction is one of the conjugated directions of the grid. This is the reason why the azimuth of the final gradient is coded on twelve directions instead of six. Each direction is coded by a numerical value taken within the interval [0,12].

3) To program the complete gradient (module and azimuth), we define first two utility procedures which generate the masks of the sups of two images and a procedure masking a greytone image by setting to any given value the pixels that fall outside the mask without modifying the others.

```
deproc masksup masksup g1 g2 b
syntax "masksup greyin1 greyin2 mask_out"
   int w ;
   w := imalloc imdepth g1
   imsub g1 g2 w
   imthresh w 1 impixmax w b
   imfree w
end

deproc masksupequal masksupequal g1 g2 b
syntax "masksupequal greyin1 greyin2 mask_out"
   int w ;
   w := imalloc imdepth g1
   imsub g1 g2 w
   imthresh w 0 0 b
   imfree w
end

deproc greymask greymask v m g
syntax "greymask val mask greyinout"
   int w ;
   w := imalloc imdepth g
   iminv m m
   immask m 0 impixmax w w
   iminf w g g
   iminv m m
   immask m 0 v w
   imsup w g g
   imfree w
end
```

Define first a rough procedure in which only the directions where the maxima of the directional gradient appear, without taking into account the fact that several directions may appear at the same point:

```
deproc vectgrad0 vectgrad0 g md az
syntax "vectgrad0 greyin module azimut(wo. corrections)"
   int i j w m gr ;
   gr := grid
   imsetgrid 1
   w := imalloc imdepth g
   m := imalloc 1
   imset 0 md
   imset 0  az
   i := 1
   for 1 to 6 do
      dirgradient i g w
      masksup w md m
      greymask 0 m az
      masksupequal w md m
      imsup w md md
      j := (i - 1)
      imcopyplane m j az
     ++ i
   end
   imfree w
   imfree m
  imsetgrid gr
end
```

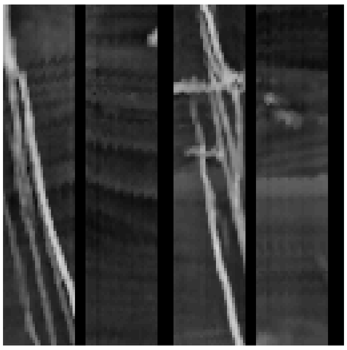
Then the vector gradient is obtained by eliminating the inconsistent directions and by taking into account the conjugated directions as already indicated. This procedure uses a programmed anamorphosis by means of the **imlookup** routine from the standard library. The anamorphosis file called **grad.lut** is a file that contains at the line i (coding of the previous directions) the coding of the new gradient direction. For example, an initial coding equal to 7 (directions 1, 2 and 3) will be equal to 3 (direction 2) after correction. Twelve directions are thus defined, the even directions correspond to the main directions of the hexagonal grid and the odd directions to the conjugated directions.

```
deproc gradvect gradvect g md az
syntax "gradvect greyin module azimut"
   int w m ;
   w := imalloc imdepth g
   m := imalloc 1
   vectgrad0 g md w
   imlookup w az "grad.lut"
   imthresh az 0 0 m
   greymask 0 m md
   imfree w
```

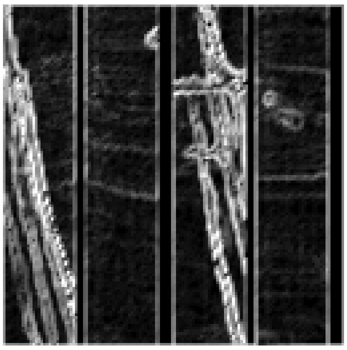
imfree m
end

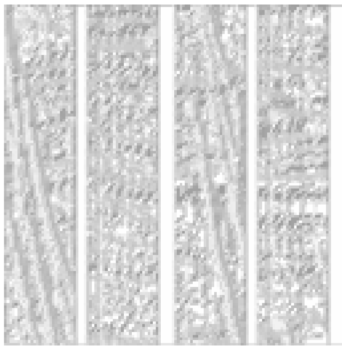
As the image of the gradient azimuths only takes thirteen values, it is not very visible. However its display may be improved either by multiplying its values, or by defining a new palette of false colors (some standard MICROMORPH functions are available for this purpose, see the help file).

The following images illustrate this gradient on the PETROLE image.



Initial image





(a) (b)

a) gradient module, b) azimuth (8x)

SUMMARY

The following transformations should now belong to your dictionary:

thinturn se1 se0 s d
thinning by rotating the structuring element se in all directions of the binary image s into the binary image d.
thin se1 se0 s d
skeleton by complete thinning by iteration of the binary image s into the binary image d.
Lthin s d
L-skeleton by thinning of the binary image s into the binary image d.
Mthin s d
M-skeleton by thinning of the binary image s into the binary image d.
Dthin s d
D-skeleton by thinning of the binary image s into the binary image d.

thickturn se1 se0 s d
thickening by rotating the structuring element se in all directions of the binary image s into the binary image d.
thick se1 se0 s d
skeleton by complete thickening by iteration of the binary image s into the binary image d.
Lthick s d
L-skeleton by thickening of the binary image s into the binary image d.
Mthick s d
M-skeleton by thickening of the binary image s into the binary image d.
Dthick s d
D-skeleton by thickening of the binary image s into the binary image d.
endpoints s d
extremities of the binary image s into the binary image d.
mulpoints s d
multiple points of the binary image s into the binary image d.
gdscentre s d
geodesic center of the binary image s into the binary image d.
gdsthinturn se1 se0 s m d
geodesic thinning by rotating the structuring element se in all directions of the binary image s into the binary image d.
gdsthin s m d
complete geodesic skeleton by iteration of the binary image s into the binary image d.
gdsthickturn se1 se0 s m d
geodesic thickening by rotating the structuring element se in all directions of the binary image s into the binary image d.
gdsthick s m d
complete geodesic skeleton by iteration of the binary image s into the binary image d.
greyseero se s d
erosion by any structuring element se defined on an elementary hexagon of the greytone image s into the greytone image d.
greysedil se s d
dilation by any structuring element se defined on an elementary hexagon of the greytone image s into the greytone image d.
greythinstep sei ses s d
thinning by any structuring element defined on an elementary hexagon of the greytone image s into the greytone image d.
greythickstep sei ses s d
thickening by any structuring element defined on an elementary hexagon of the greytone image s into the greytone image d.
gradvect g md az
gradient modulus (md) and azimuth (az) of the greytone image g.

Chapter 9

WATERSHED TRANSFORMATION

9.1. Zone of influence

We shall recall briefly what is the influence zone of the connected component of a set. Let X be a set composed of several connected sub-sets:

$$X = \bigcup_i X_i$$

The influence zone $Z(X_i)$ of the connected component X_i, is the set of the points closer to X_i than to any other connected component of X.

$$x \in Z(X_i) \Leftrightarrow d(x, X_i) < d(x, X_j), \forall j \neq i$$

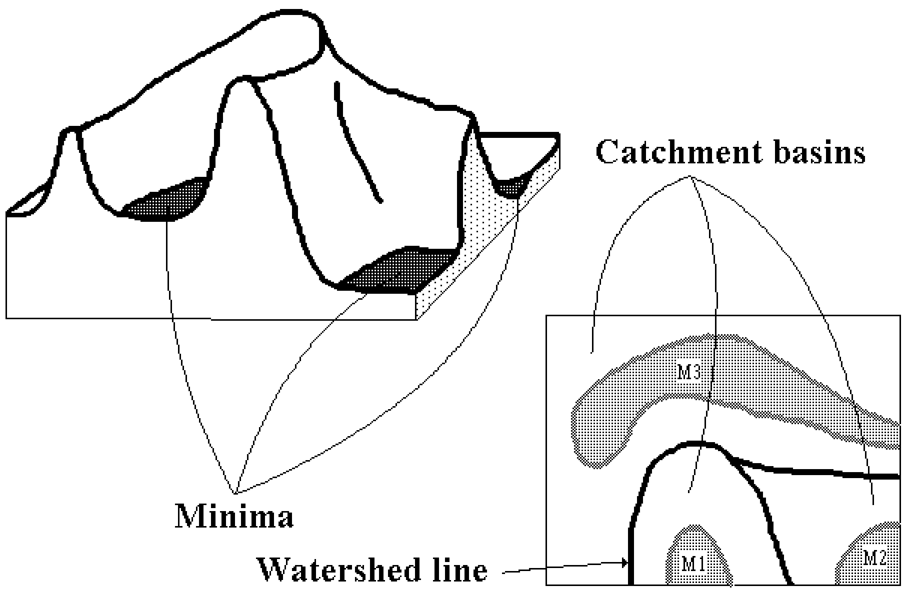
The points of the space which do not belong to any influence zone are the points of the skeleton by influence zones, or SKIZ.

9.2. Watershed transformation

Let f be a greytone image. f is assumed to take discrete values in the interval $[h_{min}, h_{max}]$. Denote $T_h(f)$ the threshold of f at level h:

$$T_h(f) = \{p, f(p) \leq h\}$$

Watershed by flooding



$Min_h(f)$ is the set of the minima of f at level h, and $IZ_A(B)$ represents the influence zone of B in A.

The set of the catchment basins of an image f is equal to the set $X_{h_{min}}$ resulting from the following recursion:

(i) $X_{h_{min}} = T_{h_{min}}(f)$

(ii) $\forall h \in [h_{min}, h_{max} - 1],\ X_{h+1} = Min_{h+1} \cup IZ_{T_{h+1}(f)}(X_h)$

The watershed of f corresponds to the complementary set of $X_{h_{max}}$, i.e. the set of the points which do not belong to any catchment basin.

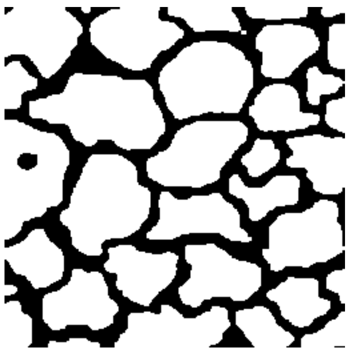
The recursion described above may be seen as a flooding process. The image f is then considered as a topographic surface in which holes are pierced in every regional minimum. The topographic surface is progressively plunged into water and dams are constructed each time that the waters coming from two distinct regional minima are on the point to merge. At the end of the flooding process the dams correspond to the watershed of f, and they delimit the catchment basins of f.

EXERCISES

Exercise n° 1 : Skeleton by influence zones

The ALUMINE image represents a polished section of alumine grains. The metallic grains are joined in the material. Since the joint thickness does not exceed a few angstroms, it is then perfectly invisible. In order to reveal it, we proceed to a chemical etching, which considerably enlarges the joints. To study the neighboring relationships between the grains, it is necessary to thin the joints.

1) Perform the SKIZ of the grains.



ALUMINE

2) Prove that the skeleton by influence zones can be obtained by a watershed with a judicious use of the distance function.

3) Compare on the COFFEE image the result obtained by this method with the result obtained with the direct transformation **skiz**.

[The SKIZ is equivalent to the watershed of the distance function]

[procedures **clip** ; **skiz**]

Solution

1) The skeleton by influence zone is performed by combining a skeleton by thickening followed by clipping.

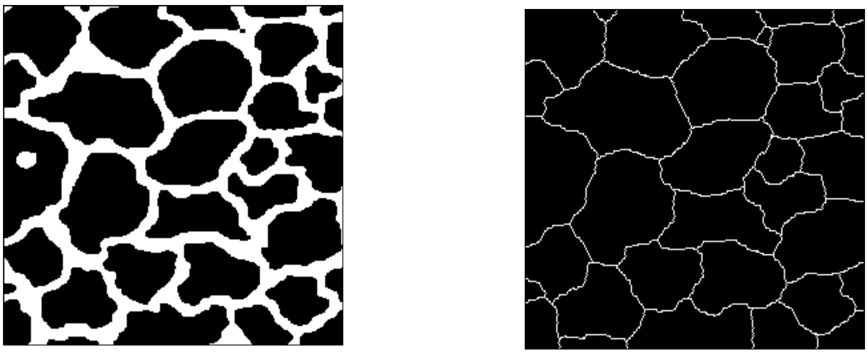
We use the procedures **Mthick** and **clip** . **clip** is so defined:

deproc clip clip s d

```
 syntax "clip binin binout"
   int se0 se1 ;
   if (grid <> 0) then
       se1 := 1 se1 := 30
   else
       se1 := 129 se0 :=62
   end
   thin se1 se0 s d
end
```

Now, we can define the word binskiz:

```
 deproc skiz skiz s d
 syntax "skiz binin binout"
   if (grid <> 0) then
      thick 2 56 s d
   else
      minidil s d
      thick 258 120 d d
   end
   iminv d d
   clip d d
end
```



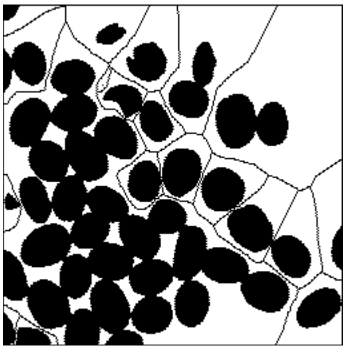
(a) (b)

Skeleton by influence zones
(a) original, (b) influence zones

2) Let X be a set. Consider the distance function of X^c . The skeleton by influence zone of X appears to be nothing but the crest lines of the function. The watershed precisely detects these crest lines. The regional minima consist of the connected components of X.

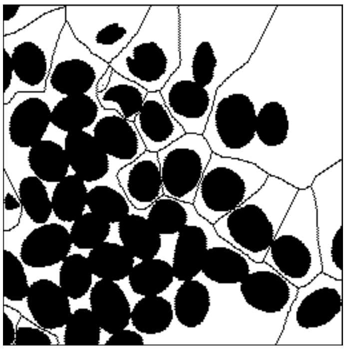
The algorithm is written thus (the initial image is assumed to be in b1):

```
iminv b1 b2
distance b2 g1
threshwshed g1 b3
```



SKIZ by watershed of the distance function of X^c

3) The "guideline" provided by the distance function in the course of successive skiz allows to avoid most digitization and order problems in the sequence of the structuring elements used for successive thickenings (this is hardly noticeable here).



SKIZ obtained by **skiz**

Exercise n° 2

Program similarly the skeleton by geodesic zone of influence.
[procedure **gdsskiz**]

Solution

The geodesic skeleton by influence zones is performed as follows:

```
deproc gdskiz gdskiz s m d
syntax "gdskiz binin binmask binout"
  int i1 i2 gr ;
  gr := grid imsetgrid 1
  imcopy s d
  i1 := imvolume d
  while (i1 <> i2) do
    gdsthickturn 2 56 d m d
    thick 30 1 d d
    imand d m d
    i2 := i1
    i1 := imvolume d
  end
```

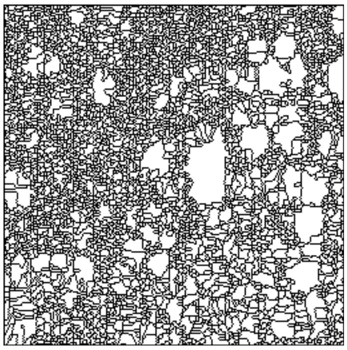
```
    imsetgrid gr
end
```

Exercise n° 3

Program the watershed described above. Apply it to the ELECTROP image.
[procedures **wshed** ; **mwshed**]

Solution

We have seen that the minima are the connected components of the threshold between 0 and N and that they cannot be reconstructed from the threshold between 0 and N-1. In order to speed up the procedure, the algorithm given below combines minima detection and flooding process (geodesic skiz).

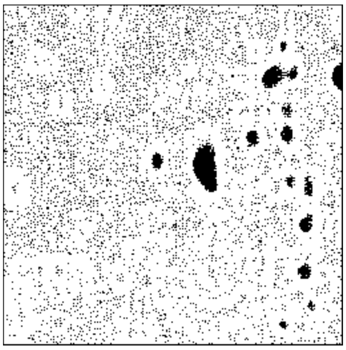


Watershed of ELECTROP image

```
deproc threshwshed threshwshed s d
syntax "threshwshed greyin binout"
  int min max SN N w sn ;
  w := imalloc 1
  SN := imalloc 1
  min := imabsmin s
  max := imabsmax s
  imset 0 d
  N := min
  for min to max do
    imthresh s 0 N SN
    gdskiz d SN d
    imcopy d w
    build SN w
    imdiff SN w w
    imsup w d d
    ++ N
  end
  imfree SN
  imfree w
end
```

This method, though it is long, has the advantage to require only little memory (only binary images are used).

The poor quality of the result may surprise you. This is a reminder that the resulting image has as many catchment basins as the regional minima contained in the initial image. Our image is very noisy and we find a great number of "parasitic" minima, as shown in the image below:



Minima of ELECTROP image

The watershed transformation can also be performed with a reduced number of thresholds. The following procedure performs this kind of operation with either a geometric or arithmetic progression of the thresholds. Note also that, in this case, the minima of the initial function are used. By these means, the final result is close to the previous one but the computation time is dramatically reduced.

```
deproc wshed wshed s d1 d2 t
syntax "wshed: greyin; binout1(divides line) binout2(minima); type(in) {t=0 for
geom.progression , t=1 for arithm. one }"
  int min max i w  k  ;
  w := imalloc 1
  min := imabsmin s
  max := imabsmax s
  minima s d2
  imcopy d2 w2
  imcopy d2 d1
  k := ( max - min )
  if ( t = 0 ) then
    i := 1
    while ( i <  k ) do
      imthresh s 0 ( min + i - 1 ) w
      imsup d2 w w
      gdskiz d1 w d1
      i := ( 2 * i )
    end
  else
    i := 0
    while ( i < k ) do
      imthresh w1 0 (min + i) w
      imsup d2 w w
      gdskiz d1 w d1
```

```
        i := i + ( k / 10 )
      end
    end
     imthresh s 0 max w
     gdskiz d1 w d1
     imfree w
  end
```

SUMMARY

The following transformation should now be part of your dictionary:

clip s d
clipping of the binary image s in the binary image d.
skiz s d
skeleton by zones of influence of the binary image s in the binary image d.
gdskiz s m d
geodesic skeleton by zones of influence of the binary image s in the binary image d.
wshed s d1 d2 type
Watershed by flooding of the greytone image s into the binary image d1. d2 contains the minima of s; type=1 corresponds to an arithmetic progress of the flooding, type=0 to a geometric one.

Chapter 10

SEGMENTATION

This chapter illustrates the use of the watershed transformation to solve some binary or greytone segmentation problems.

EXERCISES

Exercise n° 1

The two-dimensional electrophoresis is a technique for separating and identifying proteins. The migration of the proteins on the gel depends on their molecular weight and on their electric charge. From the ELECTROP image we want to extract the contour of each spot of proteins.

1) Detect the regional minima of the image. What is your conclusion? Which transformations can we apply to the image? Detect the new minima. From now on we shall work on this image.

2) Define the morphological gradient of size 1. Detect the gradient minima. Perform the watershed by means of the **wshed** procedure. Is the result satisfactory?

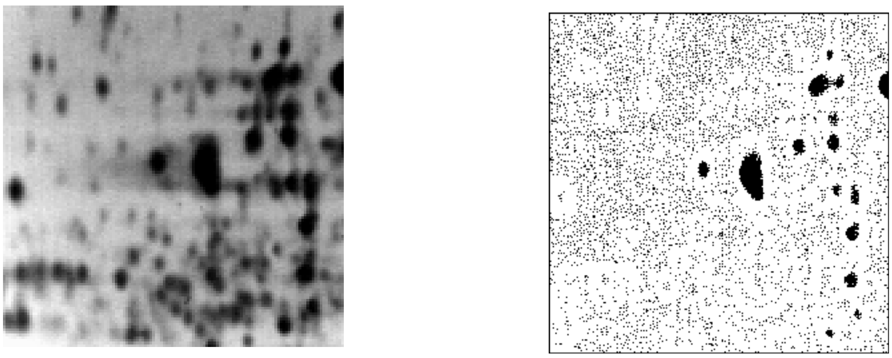
3) We will try to obtain a better result. To do so we will first seek to detect the markers located inside the spots and to contour the exterior markers as well.
- What can we take as interior markers?
- As exterior markers?
- Modify the image of the gradient so that it contains only these markers as minima (to do so use greytone reconstruction).
- Perform the watershed of this new image.

[procedure **swamping**]

Solution

1) Load ELECTROP image into g1 and do:

minima g1 b1



(a) (b)

(a) original image, (b) minima

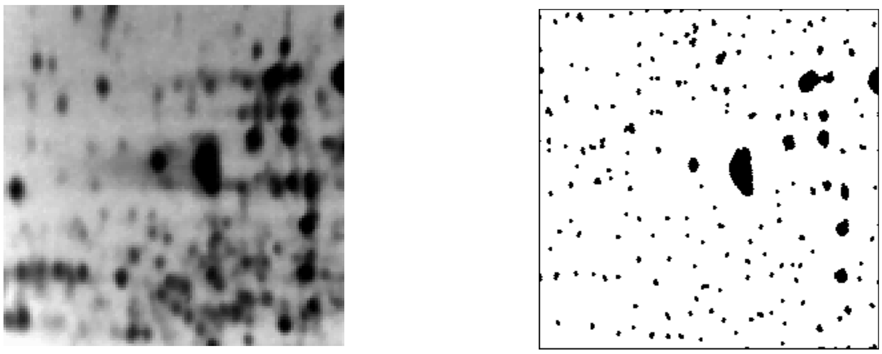
There is a great number of minima, which indicates the image is very noisy. It does not seem advisable to work directly on this image. It will be slightly filtered. An opening followed by a closing of size one are perfectly suitable in this case. Indeed, the new minima are more significant (see below).

Then, we proceed as follows:

open g1 g1 1
close g1 g1 1
minima g1 b1

2) Compute the morphological gradient of the image after filtering and detect its minima :

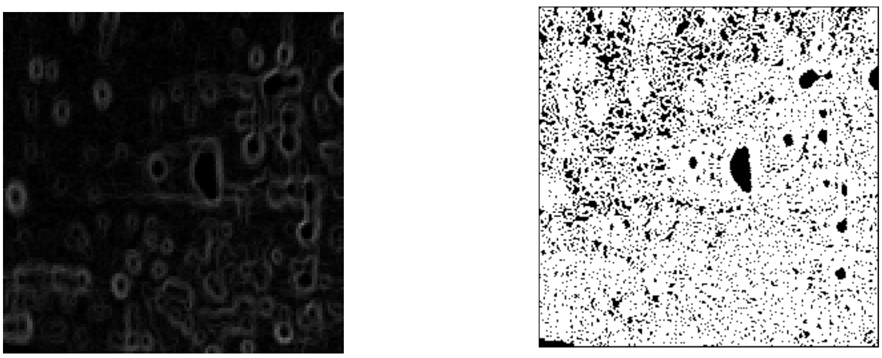
gradient g1 g2 1
minima g2 b2



(a) (b)

(a) filtered image, (b) minima

The gradient has also numerous minima. It naturally contains the minima of the image itself as well as the maxima (points of zero gradient) but also other points (points whose minimum gradient is not zero). Remember that the gradient always designates the "gradient module".



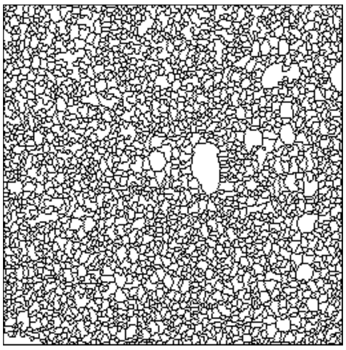
(a) (b)

(a) gradient, (b) gradient minima

Perform the watershed of the gradient image by means of the flooding procedure previously programmed.

threshwshed g2 b2

The result is very much "over-segmented" and is not completely satisfactory. However, one should not be too pessimistic. Indeed, the desired contours belong to the detected ones, even if there are too many.



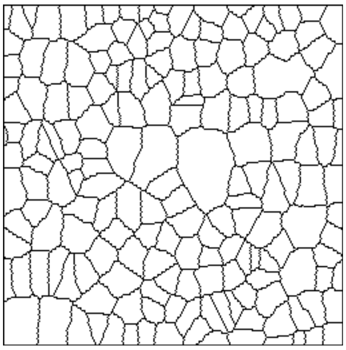
gradient watershed

3) Improving segmentation

- Our objective is that each spot is marked only once. The minima detected in the filtered image above seem to be perfectly convenient.
- In order to contour the spot correctly, we have not only to know where there are but also where there are not. Then we know that a relevant contour can be found under the form of the gradient crest lines located between the interior and exterior markers.

To do so we could use the skeleton by influence zone of the image minima:

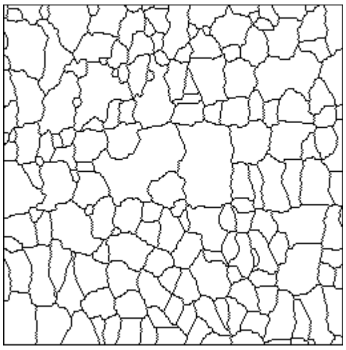
skiz b1 b2



Skeleton by influence zones of the minima

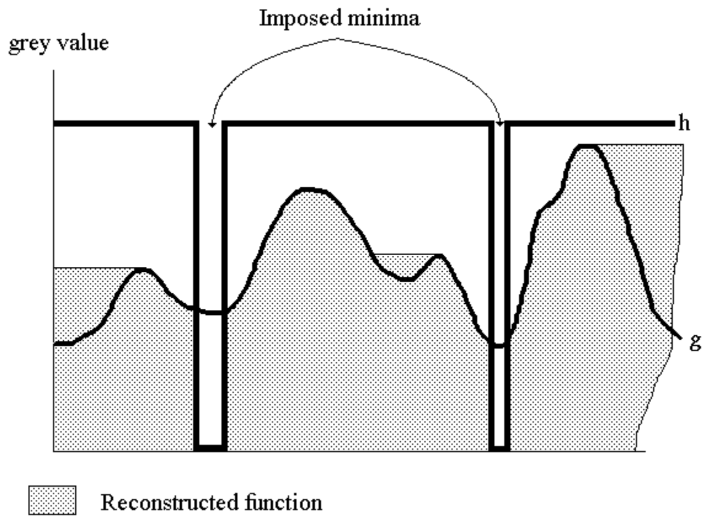
However, this does not take into account the actual topography of the image, and in particular the dimension of the spots. This is why it is preferable to perform the watershed of the image. The catchment basins replace the influence zones detected above. We are sure now that the separating lines run outside the spots.

threshwshed g1 b2



watershed of the image

- The procedure consists in constructing a "marker" image that is eroded conditionally to the gradient image as shown in the figure below.



Principle of gradient image modification

The image b1 contains the interior markers, b2 contains the exterior markers, g2 contains the gradient image.

imcopy b1 b3
imor b2 b3 b3 {union of interior and exterior markers}
iminv b3 b3
immask b3 0 255 g3

The image g3 is the mask-image used for the reconstruction.

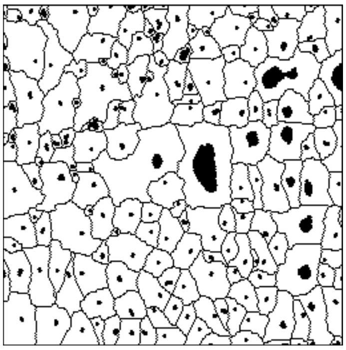
Only the fast reconstruction by geodesic dilation is installed. Therefore, we will perform this operation on the inverted images.

imcadd 1 g2 ; because the gradient contains zero minima
iminf g2 g3 g2
iminv g2 g2
iminv g3 g3

build g2 g3
iminv g2 g2
iminv g3 g3

It can be verified that the reconstructed image contains only the desired minima. To do so, apply:

minima g3 b3

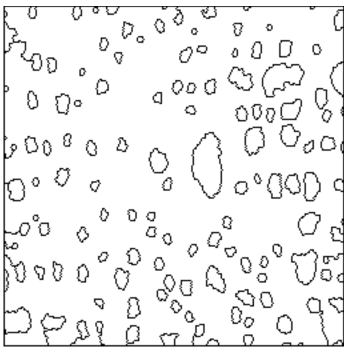


imposed minima

- The watershed of the modified gradient give the desired result.

We will use watershed by thresholds:

threshwshed g3 b4



final watershed

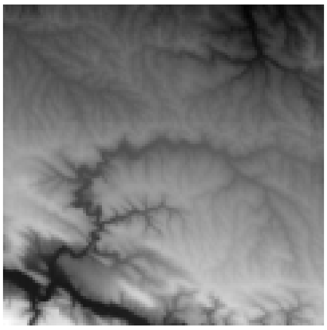
Exercise n° 2: catchment basins in a digital elevation model

This study uses the segmentation of images by watershed. It is devoted to the extraction of the catchment basins of a digital elevation model.

The RELIEF image represents a digital elevation model (DEM). In this model, the altitudes of the topographic surface are sampled at the nodes of the square grid. The purpose of this study is to define an algorithm for extracting the catchment basins from the topography. This may appear to be easy since such a transformation already exists in your tool-box. However you will see that this application is more difficult than it seems because of the noise present in the image.

1) Extract the regional minima from the RELIEF image.

2) Segment the RELIEF image into its different catchment basins and verify that to each regional minimum corresponds one and only one catchment basin.



RELIEF

The presence of undesirable regional minima within the digital elevation model induces an over-segmentation. In fact, assuming that there is no closed depression, any regional minimum should correspond to an outlet and should appear on the field border of the image.

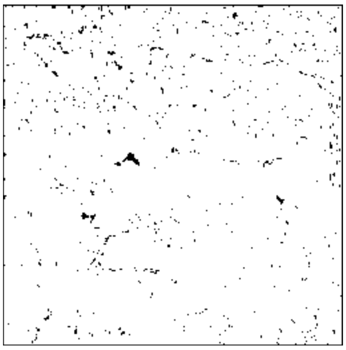
3) Indicate a procedure to suppress regional minima located inside the model. Display the mask of the pixels modified by this procedure. What are the properties of this transformation?

4) Apply the watershed transformation to the modified RELIEF.

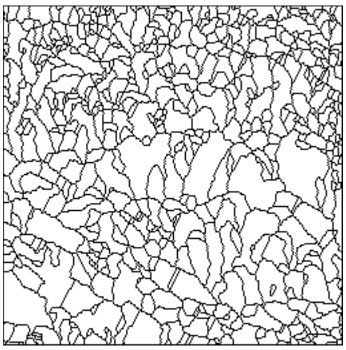
5) Propose and test a method for obtaining the catchment basin related to any point of the model.

6) What strategy would you adopt if there really existed closed depressions on the topographic surface (such as volcanic craters)?

minima of the image RELIEF



watershed of the image RELIEF



Solution

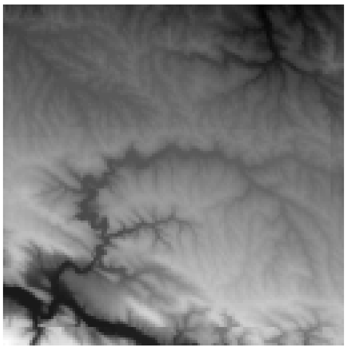
1) Here we use the minima procedure described previously.

2) The watershed procedure allows to segment the image into its different catchment basins (the image is assumed to be in b1). Note the resulting over-segmentation.

threshwshed g1 b1

3) Here we use the **clohole** procedure described in chapter 5.

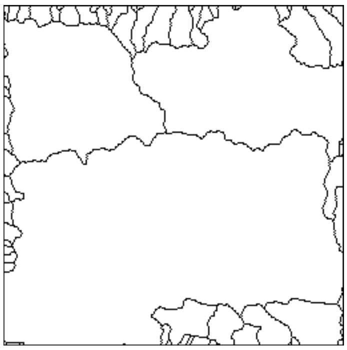
Elimination of the interior basins



This operation has all the properties of a closing: it is idempotent, increasing and extensive.

4) This time we do obtain the desired segmentation. All basins are connected with the border of the image field.

True catchment basins of the RELIEF image

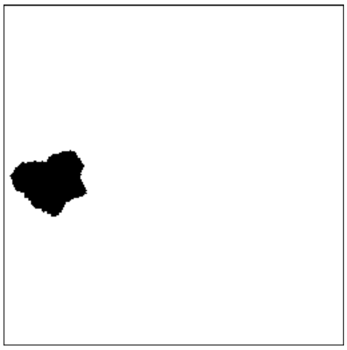


5) In order to obtain the catchment basin associated with any point, it suffices to create a local minimum at the required spot, and to run the watershed transformation once again.

Perform, for example, the following operations, the "filled" digital model is in g1.

imwritepix 0 60 140 g1
threshwshed g1 b1
imset 0 b2
imwritepix 1 60 140 b2
iminv b1 b1
build b1 b2

Catchment basin associated with a point

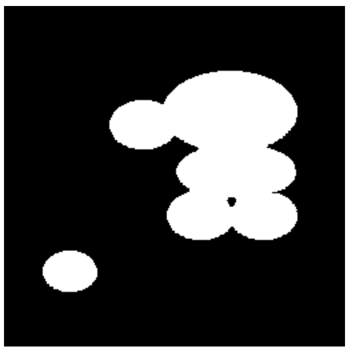


6) The watershed operator generates one and only one catchment basin for each regional minimum. If we want to preserve the catchment basins associated with depressions located inside the image, we have to work on an image in which they are still marked by a regional minimum. The filling of the holes "from the border" as described above is not suitable here. We have first to obtain, by exogenous means, the markers of the depressions we want to preserve, and then to fill holes where these markers are added to the border marker (the selection of an arbitrary point as seen in 5 is an exogenous means to obtain a marker).

Exercise n°3: separation of particles (first approach)

You have already defined (chapter 7, exercise n° 1) the ultimate erosion of the CELLS image. Using the same image, try to design an algorithm to segment the cells by means of binary operations only.

CELLS



1) Use the geodesic skeleton, or even better, the geodesic SKIZ of the ultimate eroded sets in the initial set. Is the segmentation satisfactory?

2) How can you improve this segmentation? (by taking into account the order of the ultimate eroded sets). Use again the definition of the watershed and apply it to the function:

$$f(x) = d(x,X^c)$$

- What do the thresholds Y_λ of the preceding function f correspond to?

$$Y_\lambda = \{x : d(x,X^c) > \lambda\}$$

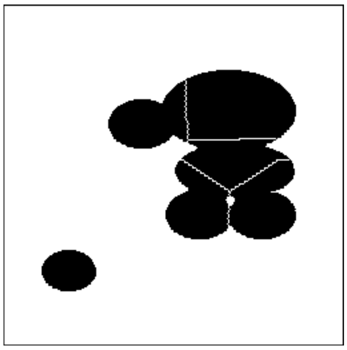
- What is the relationship between the ultimate erosion of X and the minima of f(x)?
- Program the algorithm.

3) Compare the result with that of the watershed installed in MICROMORPH (**wshed**), and with the inverted distance function (obtained by the word **distance**, also existing).

Solution

1) The skeleton by geodesic influence zone of the ultimate eroded set gives the following image (the CELLS image is in b1):

binultim b1 b2
gdskiz b2 b1 b3

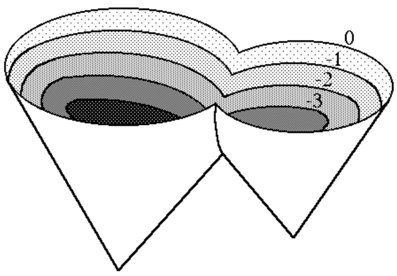


The segmentation is not very satisfactory, because the procedure does not take into account the moment the different components of the ultimate eroded set appear.

2) In fact the separating lines correspond to the watershed of the function f(x)=-d(x,X) (distance to the boundary).

The watershed of a function f can be programmed in an iterative way, by constructing step by step the set W that corresponds to the catchment basins of f. To do so, we use different thresholds X_i of the function f.

$$X_i = \{x ; f(x) < i\}$$



In the present case, the function f defined by:

$$f(x) = -d(x, X^c)$$

is either negative or zero.

Let x be a point of X such that: $f(x) \leq -i$. Then: $d(x, X^c) \geq i$. The ball of radius i centered in x then belongs to X_i and $x \in X \ominus iB$.

Let i_{max} be the smallest erosion size of X such that $X \ominus i_{max}B = \emptyset$. Then g(x): $f(x)+i_{max}$ is positive or zero and has the same watershed as f.

Let X_i be a threshold of g(x) :

$$X_i = \{x; g(x) \leq i\} = \{x; f(x) \leq i - i_{max}\}$$

Which gives:

$$X_i = X \ominus (i_{max} - i)B$$

Thus the procedure can be defined as follows:

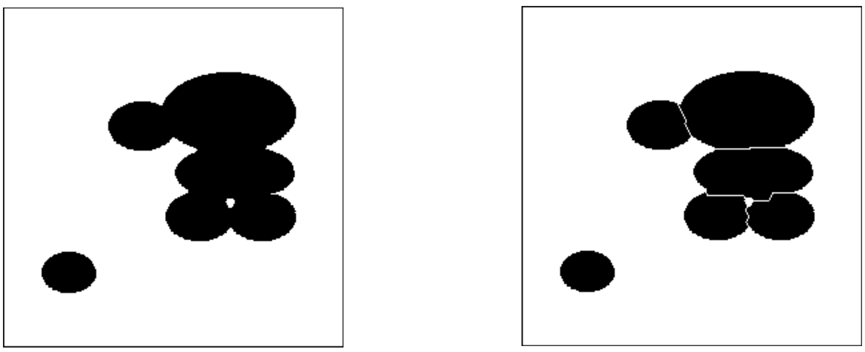
```
deproc binsegment binsegment s d
syntax "binsegment binin binout"
   int i j w1 w2;
   w1 := imalloc 1
   w2 := imalloc 1
   imcopy s d
  while imvolume d do
      ++ i
      ero d d 1
   end
   j := i
   while j do
       imcopy s w1
       ero w1 w1 -- j
       gdskiz d w1 d
       gdsdil d w1 w2 1
       imdiff w1 w2 w2
       imsup w2 d d
   end
   imfree w1
   imfree w2
end
```

Load the CELLS image into b1 and start the procedure.

(a) (b)

Segmentation by watershed, (a) original image, (b) segmentation

We can see immediately that the minima of the distance function which are added as



they appear correspond to the different connected components of the ultimate eroded set.

The segmentation applied to the CELLS image clearly shows the separations at the junctions.

However, the separations produced by digitization are not rectilinear.

3) The watershed operation installed in MICROMORPH does not proceed by geodesic thickening. Consequently it is less subjected to the hazards of digitization as can be seen on the resulting image (the CELLS image is in b1):

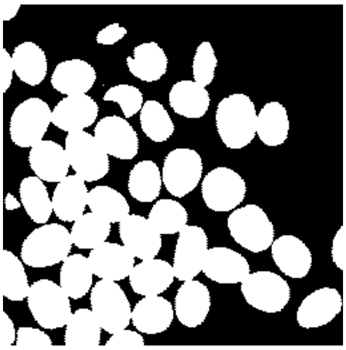
distance b1 g1
iminv g1 g1
threshwshed g1 b2
imdiff b1 b2 b3

Exercise n° 4: separation of particles (second example)

The previous case-study showed how to program watershed segmentation by means of binary operations only. It also presented an ideal case of segmentation which works immediately. This is not always the case as will prove the next example: COFFEE.

COFFEE

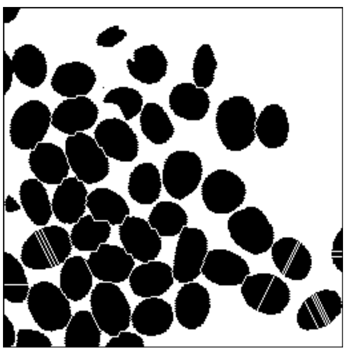
1) Apply the segmentation algorithm seen previously. What can you observe? Why?
2) Show how the filtering of the distance function allows to overcome this difficulty.



Solution

1) The direct use of the segmentation algorithm on the COFFEE image results in what is usually called an "over-segmentation". Some components which are obviously composed of only one grain of coffee are split into several connected components.

A closer examination shows that every new connected component is indeed marked by only one ultimate eroded set. The value of the segmentation is that of the eroded set. If the latter marks the same coffee grain with several connected components, this grain will finally be



cut into several pieces.

But why does the ultimate eroded set generate as many markers for one grain? This is due to the irregularities in the contour of the grains: a small concavity on the border may produce the split of the ultimate eroded set, which can also be interpreted in terms of regional maxima of the distance function.

2) The problem is how can we obtain only one marker for each coffee grain, since the ultimate erosion cannot achieve this.

We could act directly on the image itself, in order to suppress the concavities responsible for our difficulties. This method however is not simple to implement, as filling concavities is complex in the digital domain, and relevant concavities must be preserved, since they indicate the presence of two overlapping grains.

In fact it is preferable to act on the ultimate erosion itself. Even better, as the ultimate eroded set may be considered as regional maxima of the distance function, we will try to transform the distance function into a new function whose maxima will provide more appropriate grain markers.

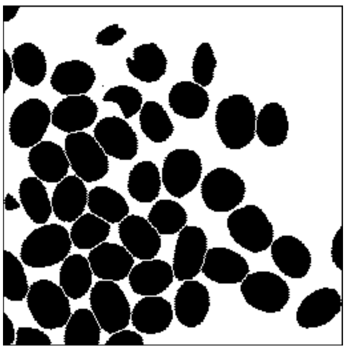
As we have already seen , the extended regional maxima give better markers. The filtering of the corresponding distance function is an opening by subtraction of a constant followed by a reconstruction.

Owing to the properties of the distance function (which is a special kind of digital image) we get the same result as would an opening by erosion-reconstruction (since the distance function of the eroded set is obtained by subtraction of the erosion size from the original distance function).

However, a simple opening of the distance function will not give the same result.

We can also perform a closing on the distance function, which comes down to applying the same closing to the binary image of the ultimate erosion.

distance b1 g1
buildopen g1 g1 3
iminv g1 g1
threshwshed g1 b2
imdiff b1 b2 b2



Chapter 11

MEASURES

11.1. Reminder

Any morphological transformation is to lead eventually to a measure on the transformed set. The main function of a transformation (or sequence of transformations) is to detect the objects to be measured: the openings revealing size distributions are an example.

Measures that satisfy good compatibility properties with translations and homothetics are not so numerous. Namely:
- area
- diameter variations
- perimeter
- number of connectivity

11.2. Measures and morphological transformations

Any measurement consists of two steps: transformation + counting of the points of the transformed image. Measuring an area is very simple as the transformation is identity, it then amounts to counting the points of the set. Besides, this measure is the only one already available in the primitives of the MICROMORPH language.

In digitized images, the intercept numbers correspond to diameter variations:

$$I_1 = N(0\ 1)\ ;\ I_2 = N\begin{pmatrix} & 1 \\ 0 & \end{pmatrix}\ ;\ I_3 = N\begin{pmatrix} 1 & \\ & 0 \end{pmatrix}$$

The transformation associated with this measure then consists in extracting the intercepts.

Measuring the number of connectivity is a little more complex, since it requires double transformations and counting:

$$v = N\begin{pmatrix} 0 & & 0 \\ & 1 & \end{pmatrix} - N\begin{pmatrix} & 0 & \\ 1 & & 1 \end{pmatrix}$$

Remind that in the plane, the number of connectivity is considered as the number of connected components minus the number of holes.

Programming measures is not enough, you have to learn how to interpret the results. Some of the following exercises are destined to familiarize you with this important step of morphological treatments.

11.3. Suggestions

In the exercises of this chapter, ratios are greatly used. These values comprised between 0 and 1 cannot be handled by MICROMORPH which recognizes only integer values. In order to overcome this difficulty, we shall use ratios expressed in percentages, integer values comprised between 0 and 1000 in the binary case and between 0 and 100 in the greytone case (cf. procedure **granul**).

EXERCISES

Exercise n° 1

1) Program the measurement of the intercept numbers in the various directions of the hexagonal grid.

2) Program the measurement of the number of connectivity. Test your algorithms on the HOLES, CELLS images, etc. .

3) Program the measurement of the perimeter. Indicate several ways of obtaining this measure and compare their respective accuracy.

[procedures **diameter** ; **cnumber** ; **digperim** ; **perim**]

Solution

1) Programming intercept numbers

```
defunc diameter diameter dir s
syntax "diameter dir binin -> diametral variation, i.e.number of intercepts"
    int w eg ;
    w := imalloc 1
    eg := edge
    imsetedge 0
    iminv s w
    iminfngb s w dir 1
   diameter := imvolume w
    imsetedge eg
    imfree w
end
```

2) Programming the connectivity number

To obtain the connectivity number, it suffices to apply the formula provided by the exercise handbook. Make sure however that the origin of the second structuring element is placed at point 1, otherwise the measure would be biased on the border of the field of analysis.

```
defunc cnumber cnumber s
syntax "cnumber binin; connectivity number of binin"
    int w eg ;
    w := imalloc 1
    eg := edge
    imsetedge 0
    oldgrid := grid
    imsetgrid 1
    imcopy s w
    imdiffngb s w 1
    imdiffngb s w 6
    cnumber := ( imvolume w )
```

```
    imcopy s w
    iminfngb  s w 2
    imdiffngb s w 1
    cnumber := (cnumber - ( imvolume w ) )
    imsetgrid oldgrid
    imsetedge eg
    imfree w
  end
```

3) Perimeter measurement

There are several ways (good or bad) of measuring the perimeter, among which:

- interior contour measurement

It is obtained by computing the number of configurations:

```
    0
  1   1
```

as well as all their rotations. Configurations like this:

```
    0
  1   1
    0
```

should be counted twice.

The procedure allowing this calculation can be written for the preceding measurements by means of the image shift primitives of MICROMORPH.

```
defunc digperim digperim s d
syntax "digperim binin binout -> digital contour length for hex. grid"
   int w  w1 w2 oldgrid i ;
  oldgrid := grid  imsetgrid 1
  w := imalloc 1
  w1 := imalloc 1
  w2 := imalloc 1
  iminv s w2
  imset 0 d
  i := 1
  for 1 to 6 do
    imcopy s w1
    iminfngb s w1 i 1
    iminfngb  w2 w1 (i mod 6 + 1) 1
   digperim := ( digperim + imvolume w1)
    imor w1 d d
    i := (i + 1)
  end
  imfree w
  imfree w1
  imfree w2
  imsetgrid oldgrid
end
```

- Exterior contour measurement

Measuring the exterior contour comes down to measuring the contour of the complementary set. Set X could be inverted, and the procedure **digperim** performed on X. However, border effects occur, since complementing is only possible within the field of analysis.

- Cauchy formula (at least, its digital version)

This formula relates the perimeter to the mean diametrical variation D in all the α directions:

$$L = \frac{1}{2}\int_{2\pi} D_\alpha d\alpha = \int_\pi D_\alpha d\alpha$$

Its digital version is given by:

$$L = \Sigma I_i$$

where I_i is the intercept number in the direction i (3 main directions).

We could just add the three intercept values given by **diameter**. However, the perimeter measurement on the complementary set would give different results due to the same border effects as those encountered previously.

In order to obtain a symmetrical measurement with identical results on X and X^c , it is sufficient to postulate that all intercepts calculated on the field border cannot be used in the perimeter measurement. However, it is then necessary to consider the intercepts of X in the six directions of the hexagonal grid. Indeed, the above-mentioned hypothesis implies there may be a different number of intercepts in direction 1 and in direction 4. The digital formula becomes:

$$L = \frac{1}{2}\sum_{i=1}^{6} I_i$$

```
defunc perim perim s
syntax "perim binin -> value: perimeter from Cauchy formula"
  int w1 w2 i perim1 perim2 eg ;
  w1 := imalloc 1
  w2 := imalloc 1
  eg := edge
  imsetedge 0
  iminv s w1
  if (grid = 0) then
    i := 1
    for 1 to 4 do
      imcopy w1 w2
      iminfngb s w2 (2 * i) 1
      perim1 := (perim1 + imvolume w2)
      i := ( i + 1)
    end
    perim1 := (( perim1 * 5) / 7)
    i := 1
    for 1 to 4 do
      imcopy w1 w2
     iminfngb s w2 ((2 * i) - 1) 1
     perim2 := (perim2 + imvolume w2)
     i := (i + 1)
   end
   perim := (((perim1 + perim2) * 11) / 28)
   else
     for 1 to 6 do
```

```
        imcopy w1 w2
        iminfngb s w2 ++ i 1
        perim :=  (perim + imvolume w2)
      end
     perim := (perim * 11 / 21)
    end
    imsetedge eg
    imfree w1
    imfree w2
  end
```

Only the last measurement is symmetrical. The first two measurements could be made symmetrical by taking their average. But it can be proved that:

$$\text{digperim}(X^c) - \text{digperim}(X) = 6n \text{ (n connectivity number)}$$

Then we have:

$$(\text{digperim}(X^c) + \text{binperim}(X))/2 = \text{digperim}(X) + 3n$$

Since this average depends on the number of connected components of the picture, it is biased. Moreover, it is not exact for the objects cutting the field border.

Exercise n° 2: transitive and stationary hypotheses

Observing the GRAINS1 image suggests that the set under study is entirely known and included in the field of measurement. In that case, it is quite legitimate to speak of the area of the set and of its number of connectivity. Similarly, it is possible to define the area of the eroded (or dilated) set provided that it is entirely contained in the field of measurement. Under this assumption of exhaustive knowledge, the working mode is said to be transitive.

Conversely, on the GRAINS2 image, such a hypothesis is hardly acceptable. Obviously, only a part of a more extended set appears in the field of analysis. In that case, the only meaningful measures are those related to the unit area: ratio, specific number of connectivity, etc. . We then speak of a stationary working mode. Measurements are performed by constructing non biased estimates. Thus, the ratio of a set X is estimated by:

$$t = \frac{\text{mes}(X \cap D)}{\text{mes}(D)}$$

D is the field of measurement, and $X \cap D$ correspond to the part of set X that is known.

1) Can you compute a non biased estimate of the ratio of the eroded set X? (As the eroded set X is biased in the field D, you have to find a field D' in which the eroded set is exact, and use it to obtain the ratio estimate).

2) Same question for the dilated set.

3) Similarly, give an unbiased estimate of the ratio of the opened and closed sets. (The figure below gives you some hints as to how you can answer the question).

[hint: edge 01, warning! you must divide by the area of the eroded field]

Solution

1) X being known in the field D only, to calculate an unbiased estimate of the ratio of the eroded set $X \ominus B$, then consists in looking for a field D' in which the eroded set is not biased.

One should have :

$$[(X \cap D) \ominus B] \cap D' = (X \ominus B) \cap D'$$

$$\text{i.e.: } (X \ominus B) \cap (D \ominus B) \cap D' = (X \ominus B) \cap D'$$

This equality is satisfied for any X if :

$$(D \ominus B) \cap D' = D'$$

Assume that $D' \supset D \ominus B$. Then:

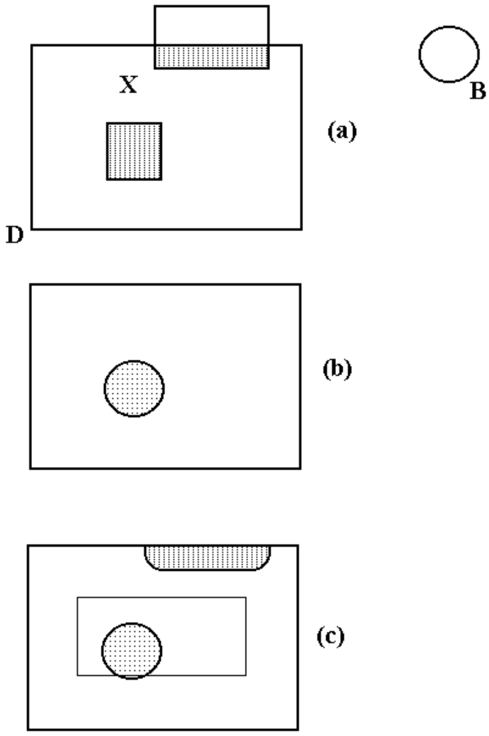
$$(D \ominus B) \cap D' = D \ominus B = D'$$

Therefore, the largest mask D' in which the eroded set is unbiased is the mask D eroded by B.

Thus, an unbiased estimate of the eroded set ratio will be:

$$t_{X \ominus B} = \frac{mes[(X \cap D) \ominus B]}{mes(D \ominus B)}$$

(a) Original image, (b) Opening obtained in the field, (c) True opening



2) If the ratio of the dilated set $X \oplus B$ is not biased, so will be the ratio of the eroded set $X^c \ominus B$ and vice versa. This means that the mask D' in which the ratio of the dilated set is measured is once again the mask $D \ominus B$.

Let us prove this a posteriori. The mask D' must satisfy :

$$[(X \cap D) \oplus B] \cap D' = (X \oplus B) \cap D'$$

Let us prove there is equality when $D' = D \ominus B$. The second expression can be written :

$$(X \oplus B) \cap D' = (X \oplus B) \cap (D \ominus B)$$
$$= ([(X \cap D) \cup (X \cap D^c)] \oplus B) \cap (D \ominus B)$$
$$= [[(X \cap D) \oplus B] \cup [(X \cap D^c) \oplus B]] \cap (D \ominus B)$$
$$= [[(X \cap D) \oplus B] \cap (D \ominus B)] \cup \underbrace{[[(X \cap D^c) \oplus B] \cap (D \ominus B)]}_{(b)}$$

(b) is empty. Indeed:

$$(X \cap D^c) \oplus B \subset D^c \oplus B = (D \ominus B)^c$$

Then, $(X \cap D)^c \oplus B$ is included in the complementary set of $(D \ominus B)$. The intersection is empty. Q.E.D.

We can write:

$$t_{X\oplus B} = \frac{mes[[(X \cap D) \oplus B] \cap (D \ominus B)]}{mes(D \ominus B)}$$

The opened set γ(X) is an erosion by B, followed by a dilation by B. Then, the opened set is not biased in the mask :

$$D \ominus \check{B} \oplus B = D \ominus (\check{B} \oplus B)$$

$$t_{\gamma(X)} = \frac{mes[\gamma(X \cap D) \cap (D \ominus \check{B} \ominus B)]}{mes(D \ominus \check{B} \ominus B)}$$

Exercise n° 3

C(l) denotes the covariance of size l, the measure of the area (transitive case) or of the ratio (stationary case) of the set X eroded by a pair of points at a distance l. In fact the transformation itself is often called covariance.

In the stationary case, the covariance is an estimate of the probability for a pair of points to be included into the set X assumed to extend to all the space.

1) Program the covariance. To do so refer to the suggestions made below. (Especially, as concerns the plotting of the curve).

2) Interpret the main features of the curve C(l), more particularly C(0), C(∞), tangent at the origin, and overall outlook of the curve.

3) Application to PARTIC1, PARTIC2 and EUTECTIC images.

Solution

1) The covariance measurements are stored in a text file.

```
deproc bincov bincov dir stat spc s1 s2  sz fname
syntax "bincov dir mode (1: intrinsic; 0 : transitive) spacing binin1 binin2 size_max
        file_name"
  int i j p1 p2 v w z c;
  w := imalloc 1
  v := imalloc 1
  z := imalloc 1
  output fname
  i := 0 j := 0
  imset impixmax z z
  imcopy s1 w
  p1 := ((1000 * (imvolume w)) / imvolume z)
  p2 := ((1000 * (imvolume s2)) / imvolume z)
  while (i < sz) do
     imcopy s2 v
     iminf w v v
    imdisplay v "bincov : v"
    c := imvolume v
    if (stat <> 0) then
      c := ((1000 * (imvolume v)) / imvolume z)
    end
    print [  i " " c ]
    for 1 to spc do
      imcopyngb w w dir 1 0
```

```
      iminfngb z z dir 1
    end
    i := (i + spc)
  end
 output ""
  if (stat <> 0) then
    print [ " asymptot "((p1 * p2) / 1000) ]
  end
  imfree v
  imfree w
  imfree z
end
```

2) As shown by the covariance definition, C(0) equals the ratio of set X.

$C(\infty)$ is the probability for a couple of points at an infinite distance from each other to be both included in X. This probability is equal to the probability of inclusion for two independent points (non correlated occurrences). That is:

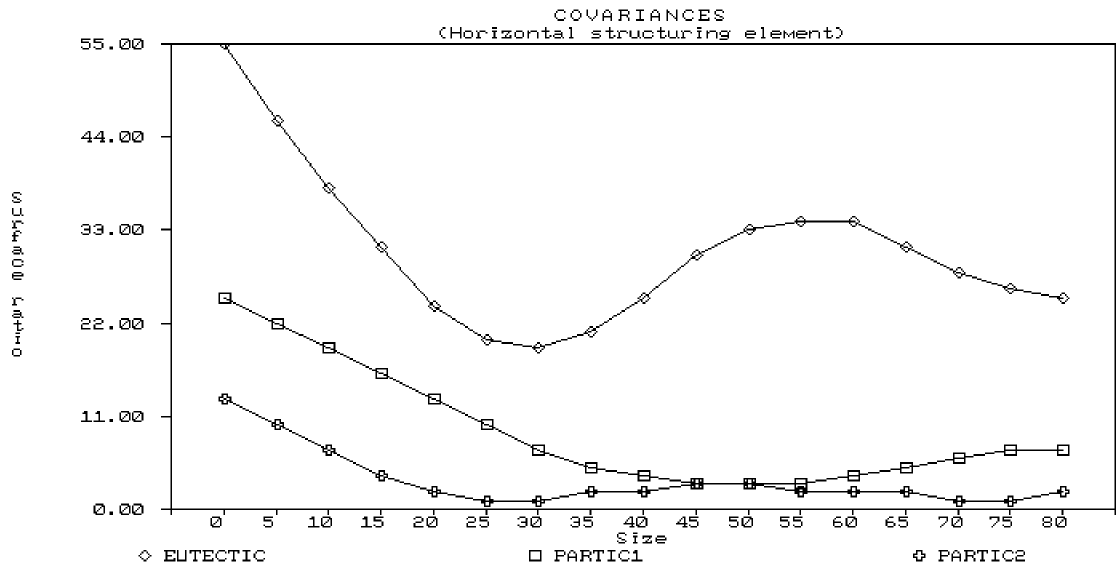
$$C(\infty) = C^2(0)$$

The tangent at the curve origin is equal in absolute value to -C'(0). An estimate of this value is given by:

$$-C'(0) = C(0) - C(1)$$

$$-C'(0) = \frac{\text{mes}(X \cap D') - \text{mes}[(X \ominus \check{K}_1) \cap D']}{\text{mes}(D')}$$

with $D' = D \ominus K_1$



Plotting of covariance curves

The numerator can be written:

$$\text{mes}[(X \cap D')/[(X \ominus \check{K}_1) \cap D']] = \text{mes}[X \cap (X \ominus \check{K}_1)^c \cap D']$$

i.e.:

$$\text{mes}(X \cap X^c_{-1} \cap D')$$

But $X \cap X^c_{-1}$ is the set X eroded by the structuring element 01, which allows to calculate the number of horizontal intercepts.

-C'(0) is then an unbiased estimate of the vertical diametrical variation of X.

3) Plotting of the curves

Note how the periodical structure of the image EUTECTIC is reflected by the covariance curve.

Exercise n° 4

The size distribution by opening has already been defined (see exercise 6, chapter 2). $F(\lambda)$ denotes the ratio of the opening of size λ of a set X.

1) Verify that:

$$G(\lambda) = \frac{F(0) - F(\lambda)}{F(0)}$$

this value is always comprised between 0 and 1. Compute it according to the area of the opened set and to the area of the eroded field of measurement D' ($D' = D \ominus 2\lambda H$).

2) Program this measurement. Application to the size distribution of the METAL1 and METAL2 images.

[procedures **granul** ; **isogranul**]

Solution

1) Prove that $G(\lambda)$ is comprised between 0 and 1.

Given:

$$G(\lambda) = 1 - \frac{F(\lambda)}{F(0)}$$

$G(\lambda)$ is monotonous, since the opening is anti-extensive, with G(0) = 0. When $\lambda \to \infty, F(\lambda) \to 0$, $G(\lambda) \to 1$.

An estimate of $G(\lambda)$ is given by:

$$G(\lambda) = \frac{mes(X \cap D') - mes(X_{\lambda B} \cap D')}{mes(X \cap D')}$$

or else:

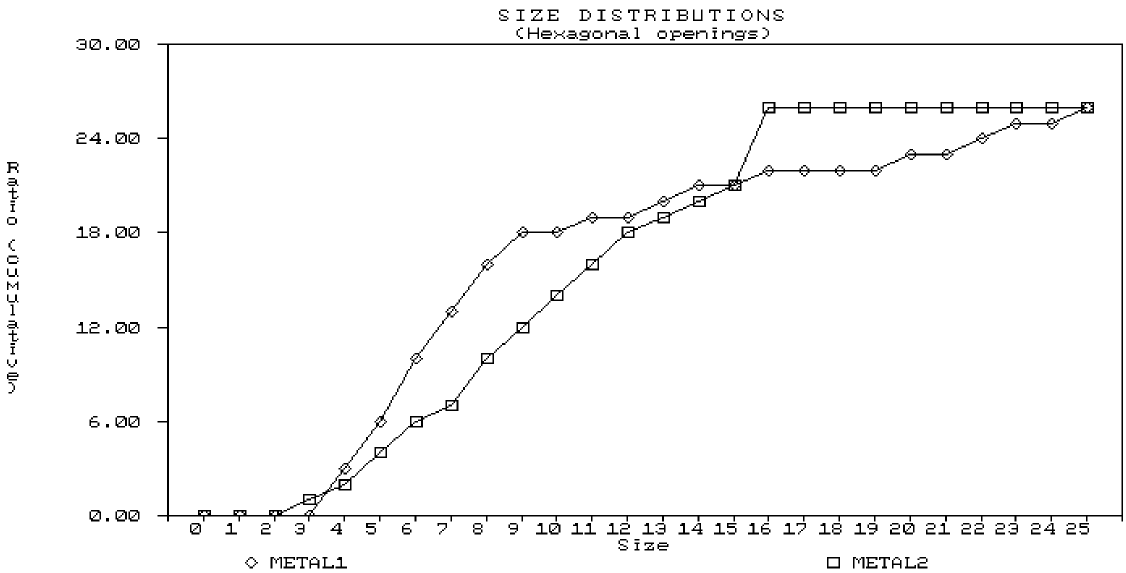
$$G(\lambda) = \frac{mes[X/X_{\lambda B} \cap D']}{mes(X \cap D')} \quad (D' = D \ominus 2\lambda B)$$

2) Computation of G(n)

The computation of G(n) is obtained as follows (binary case):

```
defunc bingranul bingranul s spc sz fname
syntax "bingranul: binin spacing size file_name"
   int z w1 w2 i a ;
   w1 := imalloc 1
   w2 := imalloc 1
   z := imalloc 1
   i := 0
   imcopy s w1
  imset impixmax z z
  imdisplay w2 cat [ "bingranul : " w2 ]
  output fname
  while (i < sz) do
     ero w1 w1 spc
```

```
      dil w1 w2 (i + spc)
      if (edge = 0)  then
         ero z z (2 * spc)
         iminf z w2 w2
        a := imvolume z
    else
      a := imvolume s
    end
    print[  (i + spc) "   " (1000 - (imvolume w2 * 1000 / a))]
    i := (i + spc)
    end
  bingranul := (1000 - (imvolume w2 * 1000 / a))
  output ""
  imfree w1
  imfree w2
  imfree z
end
```



Cumulated granulometric curves

SUMMARY

At the end of these exercises, your dictionary should be enriched with the following transformations:

diameter dir s
number of intercepts in a direction dir of the binary image s.
cnumber s tr
number of connectivity of the binary image s. if tr=1, transitive hypothesis, if tr=0, stationnary hypothesis.

digperim1 s d
perimeter of the binary image s (using interior contour), contour image in d..
perim s
perimeter of the binary image s (using Cauchy formula).
bincov dir st sp s1 s2 sz
crossed covariance of maximal size sz in a direction dir between the binary images s1 and s2, the step being sp. st=1 corresponds to the intrinsic mode, st=0 to the transitive mode.
granul s sp sz
size distribution of maximal size sz of the binary image s, step given by sp.