

MANUEL DE REFERENCE

MICROMORPH[®]



(MOTS COMPOSES)

Copyright ©1999 CMM / ENSMP / ARMINES
Tous droits réservés

Reproduction interdite sans l'autorisation de CMM / ENSMP / ARMINES. Toute représentation ou reproduction, intégrale ou partielle, faite sans le consentement de CMM / ENSMP / ARMINES est illicite (Loi du 11 Mars 1957, article 40, 1er alinéa). Cette représentation ou reproduction illicite, par quelque procédé que ce soit, constituerait une contrefaçon sanctionnée par les articles 425 et suivants du Code Pénal. La loi du 11 Mars 1957 n'autorise, aux termes des alinéas 2 et 3 de l'article 41, que les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à l'utilisation collective d'une part et, d'autre part, que les analyses et les courtes citations dans un but d'exemple ou d'illustration.

MICROMORPH est une marque déposée par ARMINES / TRANSVALOR.

MICROMORPH pour WINDOWS

Manuel de Référence des Opérations Composées

Avertissement

La structure de MICROMORPH fait que toute transformation d'image peut toujours y être considérée comme primitive pour une autre qui l'utilise. Néanmoins, on peut classer ce continuum en trois catégories pédagogiques :

- i/ les mots primitifs
- ii/ les opérations composées
- iii/ les exercices d'apprentissage de la Morphologie Mathématique.

Le manuel de référence qui suit porte sur le second niveau, et est valable pour la version WINDOWS. Les opérateurs y sont classés par thèmes. Leur liste est donnée en début de texte, par la table des procédures. Pour chaque procédure, on indique trois clés de modes, selon les conventions suivantes :

grid : 1 trame hexagonale exclusivement;
grid : 0 trame carrée exclusivement;
grid : 1,0 trame carrée ou hexagonale;
grid : [1],0 basculement automatique en carré pour la procédure, et retour à la grille antérieure (*mutatis mutandis* pour *grid* : 0, [1]).

edge : 1 mode standard exclusivement;
 : 0 mode intrinsèque exclusivement (op. croissantes) ou transitif (amincissement);
edge : [1],0 basculement automatique sur **imsetedge** = 0 le temps de la procédure, et retour à l'état antérieur (*mutatis mutandis* pour *edge* : [0], 1).

depth : bin ; grey ; bin to grey, etc... N'est pas associé à une clé, comme *grid* et *edge*, mais indique la profondeur de bits des images traitées.

Les mots primitifs qui permettent d'assigner un mode *grid* ou *edge* se nomment **imsetgrid** et **imsetegde** respectivement. Ils agissent en permanence, jusqu'à ce qu'on les modifie, ou qu' on relance micromorph. Dans ce dernier cas, c' est l'option par défaut qui est choisie, à savoir *grid* = 1, et *edge* = 1.

En fin de manuel, il est indiqué la correspondance entre la présente terminologie et celle de la version 1991, ainsi qu' un index alphabétique des procédures et des fonctions du manuel.

Table des procédures et fonctions

1 - OUTILS UTILITAIRES

clr	mise à zéro d'une image
power	(arith) fonction puissance
mod	(arith) modulo
div	(arith) division arrondie à l'entier le plus proche
abs	(arith) valeur absolue
inside, inferior	A inclus dans B, $f \leq g$
translate	translation
div2	division par deux d'une image
mean	moyenne de deux images
incrust	incruste une portion d'une image dans une autre
bintogrey	version numérique d'une image binaire
getcoords	conversion de coordonnées
masksup(equal)	masque du domaine $g_1 > g_2$, ($g_1 \geq g_2$)
greymask	constante dans un masque donné
s4rotate	rotation de 90°
greyabs	valeur absolue d'une image
format	fixe le format des images MICROMORPH
adjust	seuillage interactif
ngbnb	nombre de voisins d'un point sur la trame
dirtranspose	calcule une direction transposée
clean	mise à zéro des basses valeurs
immean	valeur de gris moyenne d'une image
getcoords	retourne les coordonnées d'un point
gandb	intersection entre une image binaire et numérique
stop	arrêt et attente de pression d'une touche du clavier
stretch	étend l'intervalle de gris d'une image
delta	masque des différences entre deux images
half	réduit d'un facteur 2 la taille d'une image

2 - OUTILS DE VISUALISATION

ds dscol	abréviations pour la visualisation
shadow shadow2	ombrage
clrcol	désactive une palette de couleurs
col	affecte une palette de couleurs
pcol	étiquetage de couleurs
cp	copie d'une image avec sa palette
cppal	copie une palette
refresh	visualise une image au-dessus des autres
profile	visualisation d'un profil de gris
coldisplay	visualisation couleur

3 - EROSIONS, DILATATIONS

3.1. Erosions et dilatations de base

dil, ero	dilatation, érosion, carrée ou hexagonale
dirdil, direro	dilatation, érosion par un segment

minidil, miniero
distance
dbldil, dblero
cont, sq4cont
gradient, sobel

dilatation, érosion, par petit carré, triangle
fonction distance carrée ou hexagonale
dilatation, érosion par doublet de points
contours internes de l'image
gradients morpho et de Sobel

3.2. Autres dilations et érosions

isodil, isoero
isodist
conedil, cyldil
crossdil
rhombodil, rhomboero
diamdil, diamero
ringdil1, ringdil2
(routines intermédiaires)

dilatation, érosion, isotrope
fonction distance isotrope
dilatation conique, cylindrique
dilatation d'un ensemble par un autre
dil., érosion, par un rhombododécaèdre
dilatation, érosion par un diamant
dilatation par un anneau hexagonal ou par ses sommets
dilatations polygonales

4 - OPERATEURS DE RANG

hexrank
median
binsegmi

opérateur de rang hexagonal
médiane (carré, hexagonal)
intersection de dil. par des segments

5 - CONVOLUTIONS

vgauss1, hgauss1, gauss1
gauss

filtres gaussiens vert., hor. et isotrope de taille 1
filtre gaussien de taille n

6 - OUVERTURES, FERMETURES

6.1. ouvertures, fermetures de base

open, close
diropen, dirclose
lineopen, lineclose
openth, closeth
lineopenth, linecloseth

ouverture, fermeture sur dil et ero
ouverture, fermeture sur dirdil et direro
réunion, (intersection) des précédentes
top hats sur open, close
top hats sur lineopen, sur lineclose

6.2 Autres ouvertures et fermetures

isopen, isoclose
infopen, supclose

miniopen, miniclose
stopen

ouverture, fermeture, isotrope
limite d'inf d'ouvertures directionnelles
(ou de sup de fermetures)
ouverture, fermeture sur minidil, miniero
inf d'ouvertures directionnelles

7 - GEODESIE ET CONNEXITE

gdsdil, gdsero
build
buildopen, buildclose
areaopen
ringbuild
gdsdist
recons
levelling

dilatation, érosion, géodésique
reconstruction binaire par marqueur
ouverture, fermeture, par reconstruction
ouverture selon l'aire
reconstruction par un anneau
distance géodésique
reconstruction binaire (similaire à build)
nivellement

8 - APPLICATIONS DE LA GEODESIE

edgeoff	supprime les grains touchant le bord
fgrain	extrait le premier grain
border	contours internes du champ
clohole	bouche les pores internes
maxima, minima	extrema d'une fonction
extmaxima, extminima	extremas étendus
swamping	correction d'homotopie par marqueurs
extrema	extrema d'une image dans un masque
grainclose	lissage et fermeture de grains
indivaf	fermeture individuelle de particules
dynamics	dynamique des maxima et minima d'une image

9 - FILTRES

9.1. Filtres principaux

af	filtre alterné
asf, fullasf	filtre alterné séquentiel (2 types de progressions)
lineaf, lineasf	filtre alterné, alterné séquentiel sur lineopen
automed, centre	opérateur automédian, centre morpho.
contrast	contraste sur isoero-isodil, sur isopen-isoclose
contrasth	contraste sur top hat

9.2. Filtres supplémentaires

minifilt	mini filtre alterné
isaf, isasf	filtre alterné, alterné séquentiel, isotrope
buildaf, buildasf	filtre alterné, alterné séquentiel, par reconst.
cloropen	toggle entre ouverture et fermeture
minibuildaf	filtre par reconstruction avec miniopen et miniclose
binmiddle	ensemble médian binaire
grmiddle	ensemble médian numérique

10. BOULES MAXIMALES ET SQUELETTE

binopenskel	squelette par carrés (hexa.) maximaux
binultim	érodé ultime
centroid	centroïde (dernier érodé)
condbis	bissectrice conditionnelle

11 - AMINCISSEMENTS, EPAISSISSEMENTS

11.1. Amincissements, épaisissements binaires

(gds)thickturn, (gds)thinturn	cycle d'épaissements, d'amincissements, élémentaires (géodésiques)
(gds)thick, (gds)thin	épaissement, amincissement, limite
Lthick, Mthick, Dthick	épaissements homotopiques
Lthin, Mthin, Dthin	amincissements homotopiques
endpoints	points extrêmes
multpoints	points multiples
gdscentre	centre géodésique

11.2. Amincissements, épaisissements numériques

greysero, greysedil	érosion dilatation par se
greythickstep, greythinstep	épaisissement, amincissement, élémentaires
dirgradient	gradient directionnel
vectgrad0	gradient complet brut
gradvect	gradient complet élaboré
mainthin	amincissement et ébarbulage
greyhmt	transformation en tout-ou-rien numérique
greythickturn	épaisissement numérique élémentaire
greythick	épaisissement numérique

12 - LIGNE DE PARTAGE DES EAUX

clip	ébarbulage
(gds) skiz	squelette (géodésique) par zones d'influence
threshwshed	L.P.E. par seuillage
mwshed, wshed	L.P.E. marquée, ou non

13 - SEGMENTATION

mgradwshed, gradwshed	L.P.E. marquée ou non, du gradient
shapeseq	segmentation par fonction distance
smoothcnc, jumpcnc, jump	extraction de contraste
mosaic	image mosaïque
wfall	transformation hiérarchique (cascades)
kheops	pyramide d'images mosaïque

14 - MESURES

14.1. Mesures de base

diameter	variation diamétrale
digperim, perim	périmètres digital et euclidien
cnumber	nombre de connexité
binh(v)ferret	diamètre de Ferret

14.2. Mesures supplémentaires

var0, var1, var2	variances d'ordre 0, 1, 2
rug	rugosité
mse	différence quadratique moyenne
bincount	nombre de composantes connexes
flatcount	nombre de zones plates
flatzone	aire des zones plates

15 - COURBES

bincov	covariance binaire (directe ou rectangle)
vario1, vario2	variogrammes d'ordre 1 et 2
modcont	module de continuité
isogranul, granul	granulométries
covar	covariance
pvonl	mesure de l'érosion linéaire binaire
p1vonl	granulométrie linéaire binaire en nombre
p2vonl	granulométrie linéaire binaire en mesure

16 - SIMULATIONS ALEATOIRES

16.1. Simulations booléennes

point, points, regpoints
isobool, isobool2
elbool, tribool, dropbool
rose, hard, disjoint, flake
conebool, conebool2
rocky

points aléatoires
simulations d'ensembles booléens isotropes
simulations ensembles booléens anisotropes
simulations de hiérarchies
simulations de cônes booléens
fonds rocheux

16.2. Droites et partitions

lines, diags
nestlines
steps

droites poissonniennes
hiérarchies de droites poissonniennes
bandes poissonniennes

17. GRAPHERS

Gero, Gdil
Gopen, Gclose
setborder
Gbuild
label

érosion, dilatations sur un graphe
ouverture, fermeture sur un graphe
marque les bords d'une image numérique
reconstruction sur un graphe planaire
génération d'une image label

18. UTILITAIRES 3D

Seqload
Seqsave
Seqclr
Seqset
Seqcopy
Seqinv
Seqadd
Seqcadd
Seqseqadd
Seqcsub
Seqseqsub
Seqcmul
Seqmean
Seqseqmean
Seqdiff
Seqvolume
Seqinf
Seqseqinf
Seqsup
Seqseqsup
Seqbuild
Seqthresh
Seqgrad
Seqhist
Scroll, Scroll2, Scroll3
Visu

chargement d'une séquence
sauvegarde d'une séquence
effacement d'une séquence
affecte une valeur donnée à une séquence
copie une séquence
inverse une séquence
additionne toutes les images d'une séquence
additionne une valeur constante à une séquence
addition de deux séquences
soustraction d'une valeur constante à une séquence
soustraction de deux séquences
multiplie une constante à une séquence
valeur moyenne des images d'une séquence
séquence moyenne de deux séquences
module de la différence de deux séquences
volume d'une séquence
inf de toutes les images d'une séquence
inf entre deux séquences
sup de toutes les images d'une séquence
sup de deux séquences
reconstruit une séquence à l'aide d'une seq. marqueur
seuillage à niveaux de gris d'une séquence
gradient d'une séquence
histogramme d'une séquence
balayage de séquence
visualisation en perspective

19. PROCEDURES 3D

ero3D, dil3D	érosion et dilatation 3D
open3D, Close3D	ouverture, fermeture 3D
gradient3D	gradient 3D
build3D	reconstruction 3D
buildopen3D, buildclose3D	ouverture, fermeture par reconstruction 3D
openth3D, closeth3D	chapeau haut-de-forme 3D
maxima3D, minima3D	extrema 3D
af3D, asf3D	filtres alternés séquentiels 3D
buildaf3D, buildasf3D	filtres alternés séquentiels par reconstruction 3D
timdil, timero	érosion, dilatation linéaire suivant l'axe temporel
timopen, timclose	ouverture, fermeture linéaire selon l'axe temporel

20. SOURIS

fill	dessine et remplit un polygone
dline	dessine des lignes
delobj	supprime des composantes connexes
pointinfo	coordonnées et valeur de gris d'un point
binpointer	pointe et extrait une particule
objinfo	aire d'un objet pointé
brush	fonction brosse

CORRESPONDANCE
entre les opérateurs MICROMORPH versions MSDOS et Windows

	Version MSDOS		Version Windows	
	Noms		Noms	chapitres changements
Ch. 2	Nbinlinero, Ngreylinero		direro	2-1 +
	Nbinlindil, Ngreylindil		dirdil	2-1 +
	binerode, greyerode		ero	2-1 +
	Nbinerode, Ngreyerode		ero	2-1 +
	bindilate, greydilate		dil	2-1 +
	Nbindilate, Ngreydilate		dil	2-1 +
	bintriero, greytriero		miniero	2-1 +
	bintridil, greytridil		minidil	2-1 +
	binb1dil, greyb1dil		id, id	2-2 0
	binb2dil, greyb2dil		id, id	2-2 0
	bindbero, greydbero		id, id	2-2 0
	bindbldil, greydbldil		id, id	2-2 0
Ch. 3	Nbingen, Ngreyopen		open	2-3 +
	Nbiclose, Ngreyclose		close	2-3 +
	Nbinlinopen, Ngreylineopen		diropen	2-3 +
	Nbinlinclose, Ngreylineclose		dirclose	2-3 +
	Ngreysupopen		lineopen	2-3 +
	Ngreyinfclose		lineclose	2-3 +
Ch. 4	bininterc		diameter	11-1 +
	binnumber		cnumber	11-1 +
	binperim1		digperim	11-1 +
	binperim		perim	11-1 +
	binperim2		<i>supprimé</i>	
	Nbincovalc		bincov	11-4 ++
	Nbingranul		granul	11-4 ++
	binbuild		id(= fast build)	4-2 0
	binvbuild		<i>supprimé</i>	
Ch. 5	Nwtophat		openth	3-1 +
	Nbtophat		closeth	3-1 +
	NSupopen th		lineopenth	3-1 +
	Ninfcloseth		linecloseth	3-1 +
	sobel		sobel	3-1 0
	Ngradient		gradient	3-1 +
	Npreggrad, Nnpreggrad		<i>supprimé</i>	
Ch.6	bingdsdil, Nbingdsdil		gdsdil	4-1 +
	greygdsdil, Ngreygdsdil		gdsdil	4-1 +
	bingdsero, Nbingdsdil		gdsero	4-1 +

greygdsero, Ngreygdsero	gdsero	4-1	0
binborder	border	3-1	+
binedgeoff	edgeoff	5-1	
bincloholes, greycloholes	cloholes	5-1	+
binindiana	areaopen	4-2	+
Ch. 7			
binultim	id	5-1	+
binopenskel	id	3-2	+
maxima, minima	id, id	5-2	+
extmaxima, extminima	id, id	5-2	+
erodistance	isodist	2-2	++
Nbbuildopen, Ngbuildopen	buildopen	4-2	+
Ngbuildopen, Ngbuildclose	buildclose	4-2	+
threshmax	<i>supprimé</i>		
Ch. 8			
greyseerode, greysedilate	greyseero, greysedil	7-2	0
greythick, greythin	greythickstep, greythinstep	7-2	0
bincontour	cont	3-1	+
dirgradient, vectgrad0	id, id	7-2	0
gradvect, gradpalette	id, id	7-2	0
masksup(equal), greymask	id, id	1	0
bin(gds) thin, bin(gds)thick, sedual	<i>supprimés</i>		
Ch. 9			
binsk1turn, binsk2turn	thinturn, thickturn	7-1	++
binskel1, binskel2	thin, thick	7-1	++
binLskel1, binMskel1, binDskel1	Lthin, Mthin, Dthin	7-1	+
binLskel2, binMskel2, binDskel2	Lthick, Mthick, Dthick	7-1	+
binendpoints	endpoints	7-1	+
binmulpoints	mulpoints	7-1	+
bingdscenter	gdscentre	7-1	+
bingsk1turn, bingsk2turn	gdsthinturn, gdsthickturn	7-1	+
bingdskel1, bingdskel2	gdsthin, gdsthick	7-1	+
binclipall	clip	7-1	++
bin(gds)skiz	(gds)skiz	7-1	+
binhferret	id	11-2	0
binvferret	<i>supprimé</i>		
Ch.10 - Ch.11 -Ch.12			
Threshwshed	id	9-1	0
Nedcontrast, Noccontrast, Noc5contrast	contrast	6-2	+
Nthcontrast	contrasth	6-2	0
wasf, basf	fullasf	6-1	+
wlinasf, blinasf	lineasf	6-1	+
wbuildasf, bbuildasf	buildasf	6-1	
automed	automed	6-2	++
iterautomed	<i>supprimé</i>		

1. OUTILS UTILITAIRES

PROCEDURE **clr**

SYNTAXE **clr** *imin*

OBJET efface l' image *imin*

MODES *grid : 1 , 0* *edge : 1 , 0* *depth : 1 , 8 , 16*

FONCTION **inside**

SYNTAXE $v := (\text{binin1 } \mathbf{inside} \text{ binin2})$

OBJET **inside** retourne 1 si *binin1* est inclus dans *binin2*, et 0 dans le cas contraire

MODES *grid : 1 , 0* *edge : 1 , 0* *depth : 1*

VOIR AUSSI **inferior**

PROCEDURE **clean**

SYNTAXE **clean** *greyin* *threshold* *greyout*

SYNTAXE Aux points où la valeur de *greyin* est inférieure ou égale à *threshold*, on remplace cette valeur par le minimum.

VOIR AUSSI *grid : 1 , 0* *edge : 1 , 0* *depth : 8 , 16*

FONCTION **inferior**

SYNTAXE *greyin1* **inferior** *greyin2*

OBJET **inferior** retourne 1 si *greyin1* est inférieur ou égal à *greyin2*, et 0 dans le cas contraire.

MODES *grid : 1 , 0* *edge : 1 , 0* *depth : 8 , 16*

VOIR AUSSI **inside.**

PROCEDURE **div2**

SYNTAXE **div2** *greyin* *greyout*

OBJET Place dans *greyout* l' image *greyin* divisée par 2.

MODES *grid : 1 , 0* *edge : 1 , 0* *depth : 8 , 16*

FONCTION **div**
SYNTAXE $v := (a \text{ div } b)$
OBJET Calcul la division arrondie à l'entier le plus proche de a par b .
VOIR AUSSI / (mots primitifs).

PROCEDURE **mean**
SYNTAXE **mean** greyin1 greyin2 greyout
OBJET Place dans *greyout* la moyenne arithmétique de *greyin1* et *greyin2*.
MODES *grid* : 1 , 0 *edge* : 1 , 0 *depth* : 8, 16

FONCTION **immean**
SYNTAXE $v := \text{immean } \text{imin}$
OBJET Calcule la valeur moyenne des pixels de l'image *imin*.
MODES *grid* : 1 , 0 *edge* : 1 , 0 *depth* : 8

PROCEDURE **incrust**
SYNTAXE **incrust** greyin binmaskin greyinout
OBJET Impose *greyin* dans la zone *binmaskin* de *greyinout*.
MODES *grid* : 1 , 0 *edge* : 1 , 0 *depth* : 1, et 8, 16

PROCEDURE **bintogrey**
SYNTAXE **bintogrey** binin greyout
OBJET Créé dans *greyout* une image numérique. Abbréviation pour
"**immask** binin **impixmin** greyout **impixmax** greyout greyout "
où : "**impixmin** greyout" = 0 ou - 32767
et où : "**impixmax** greyout" = 255 ou 32767
MODES *grid* : 1 , 0 *edge* : 1 , 0 *depth* : 1, et 8, 16

PROCEDURE **translate**

SYNTAXE **translate** *imin imout signx x signy y*

OBJET Place dans *imout* le traduité de *imin* par le vecteur (*signx x ; signy y*). L'extérieur de *imin* est égal à 0. Par convention , on a :

droite : *signx = 3* haut: *signy = 1* gauche : *signx = 7* bas : *signy = 5*

MODES *grid : 1 , 0* *edge : 1 , 0* *depth : 1*

REMARQUE Que l'image soit en carré ou en hexagonal, le vecteur translation est défini, lui, en trame carrée.

FONCTION **s4rotate**

SYNTAXE *v := s4rotate structuring_element*

OBJET Restitue le code de l'élément structurant après rotation de 90 degrés.

MODES *grid : 1 , 0* *edge : 1 , 0* *depth : 1*

REMARQUE Noter la différence avec les 8 rotations de la procédure voisine **serotate** (en trame carrée); ici, il n'y en a que 4 .

VOIR AUSSI **serotate.**

FONCTION **power**

SYNTAXE *v := a power b*

OBJET Calcule a^b (*a* et *b* sont des valeurs entières).

FONCTION **mod**

SYNTAXE *v := a mod b*

OBJET Calcule *a* modulo *b*.

PROCEDURE **immul**

SYNTAXE **immul** *imin1 imin2 imout*

OBJET Multiplie *imin1* par *imin2*, pixel à pixel, et place le résultat dans *imout*. *Imout* doit être une image 16-bit.

MODES *grid : 1 , 0* *edge : 1 , 0* *depth : 8, 16*

PROCEDURE **masksup**
SYNTAXE **masksup** greyin1 greyin2 binout
OBJET En *binout*, ensemble des points où *greyin1* est supérieur à *greyin2*
MODES *grid : 1, 0* *edge : 1, 0* *depth : 8, 16, et 1*
VOIR AUSSI **masksupequal**.

PROCEDURE **masksupequal**
SYNTAXE **masksupequal** greyin1 greyin2 binout
OBJET En *binout*, ensemble des points où *greyin1* est supérieur ou égal à *greyin2*.
MODES *grid : 1, 0* *edge : 1, 0* *depth : 8, 16 et 1*

PROCEDURE **greymask**
SYNTAXE **greymask** entier binin greyout
OBJET Engendre dans *greyout* l' image numérique "*entier* binin*".
MODES *grid : 1, 0* *edge : 1, 0* *depth : 1 et 8, 16*
REMARQUE Il faut que l'on ait $1 \leq \text{entier} \leq 255$ (ou 32767) .

FONCTION **abs**
SYNTAXE $v := \text{abs } a$
OBJET Renvoie la valeur absolue de *a*.

PROCEDURE **greyabs**
SYNTAXE **greyabs** greyin greyout
OBJET Routine produisant la valeur absolue d'une image numérique. L'origine est à la cote 127 pour les images de profondeur 8 bits, et à 0 pour celles de 16 bits.
MODES *grid : 1, 0* *edge : 1, 0* *depth : 8, 16*

PROCEDURE **format**

SYNTAXE **format** vertical horizontal

OBJET Initialise le nouveau format d' image (vertical x horizontal), détruit toutes les images précédentes, et alloue 10 images de 1 bit: b1.. b10, 10 images de 8 bits: g1... g10, et 3 images de 16 bits: h1.. h3.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

FONCTION **ngbnb**

SYNTAXE v := **ngbnb**

OBJET Retourne le nombre de voisins dans la grille courante.

VOIR AUSSI **grid**

FONCTION **dirtranspose**

SYNTAXE v: = **dirtranspose** direction

OBJET Calcule la direction transposée de la *direction* entrée.

PROCEDURE **adjust**

SYNTAXE **adjust** image_gris image_bin seuil_bas seuil_haut

OBJET Permet d'effectuer un seuillage interactif de l'image d'entrée à l'aide du pavé numérique entre les niveaux de seuil définis par *seuil_bas* et *seuil_haut*. Les seuils de départ sont *seuil_bas* et *seuil_haut*. Il est ensuite possible de modifier ces valeurs en utilisant les touches 0, 1, 2, 3, 7, 8 et 9 du pavé numérique.

0 : fin du seuillage interactif
7 :incrémentation de N de *seuil_bas*
1 : décrémentation de N de *seuil_bas*
9 :incrémentation de N de *seuil_haut*
3 : décrémentation de N de *seuil_haut*
8 :incrémentation de 1 de N
2 : décrémentation de 1 de N
Au départ, N vaut 1.

VOIR AUSSI **imthresh**

PROCEDURE **getcoords**
SYNTAXE **getcoords** *fpoint_code* x y
OBJET Convertit le code *fpoint_code* retourné par la fonction *fpoint* ou *imcompare* en coordonnées *x* et *y* du point.
VOIR AUSSI **imcompare**, **fpoint**

PROCEDURE **gandb**
SYNTAXE **gandb** *greyin* *binin* *greyout*
OBJET Restreint l'image *greyin* au masque *binin* et place le résultat dans *greyout*.
MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

PROCEDURE **stop**
SYNTAXE **stop**
OBJET Emet un son pour indiquer l'arrêt et attend la pression d'une touche au clavier pour continuer l'exécution du programme.
VOIR AUSSI **readkey**

PROCEDURE **stretch**
SYNTAXE **stretch** *imin* *imout*
OBJET Etend les niveaux de gris de *imin* sur tout l'intervalle 0-255.
MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 8

PROCEDURE **delta**
SYNTAXE **delta** *im1* *im2* *binout*
OBJET Les deux entrées *im1* et *im2* peuvent être binaires ou numériques. La procédure place en *binout* l'ensemble des points où $im1 \triangleleft im2$ et imprime le volume de $|im1 - im2|$ si *im1* et *im2* sont à niveaux de gris, ou l'aire de leur différence symétrique s'ils sont binaires.
MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1,8

PROCEDURE half

SYNTAXE half imin imout type

OBJET Divise par deux la taille de l'image *imin* et place le résultat dans *imout*.

Si *type* = 0 , l'image *imin* est réduite directement,

Si *type* = 1 , l'érosion miniero est effectuée avant la réduction,

Si *type* = 2 , l'ouverture miniopen , suivie de la fermeture miniclose
sont effectuées avant la réduction,

L'image réduite occupe le coin en bas à gauche de *imout*.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1,8

REMARQUE Quand on veut travailler avec des images deux fois plus petites, il faut d'abord appliquer la procédure half sur les images d'étude, puis format pour réduire la taille des fenêtres. Les images réduites seront alors correctement cadrées dans les nouvelles fenêtres.

2. OUTILS DE VISUALISATION

PROCEDURE **ds**

SYNTAXE **ds** *imin*

OBJET Visualise l'image *imin*. Lorsque cette procédure est utilisée à l'intérieur d'une autre procédure (pour visualiser par exemple une mémoire de travail), le numéro de la mémoire est affiché ainsi que la profondeur de la routine appelant la visualisation.

MODES *grid* : 1 , 0 *edge* : 1 , 0 *depth* : 1 , 8 , 16

PROCEDURE **dscol**

SYNTAXE **dscol** *imin1 imin2 zoom*

OBJET Si *imin1* et *imin2* sont binaires, la procédure visualise la première en rouge, et la seconde en vert. Si l'une des deux seulement est binaire, elle est visualisée en rouge, et se superpose à l'image numérique sans la masquer.

MODES *grid* : 1 , 0 *edge* : 1 , 0 *depth* : 1 , 8

REMARQUE Il est nécessaire qu' au moins une des deux images à visualiser soit binaire. La visualisation de toutes les autres images reste gelée pendant que la procédure est active.

VOIR AUSSI **coldisplay** - version étendue de **dscol**

PROCEDURE **shadow**

SYNTAXE **shadow** *greyin greyout*

OBJET Ombrage permettant de visualiser les images numériques comme des reliefs. Fonctionne d'autant mieux que les pentes sont proches de l'unité. Spécialement indiqué pour les fonctions *distance* .

MODES *grid* : [1] , 0 *edge* : 1 , 0 *depth* : 8 , 16

VARIANTE **shadow2** (direction d'ombrage différente).

PROCEDURE **clrcol**

SYNTAXE **clrcol** *imin*

OBJET Désactive la palette de couleur affectée à l'image *imin*. Quand *in* est remplacé par 0, toutes les images couleur redeviennent grises.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8*

VOIR AUSSI **col, pscol, dscol**

PROCEDURE **col**

SYNTAXE [v :=] **col** *imin pal*

OBJET Affecte la palette *pal* (nom de fichier) à l'image *imin*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8*

VOIR AUSSI **clrcol, pscol, dscol**

PROCEDURE **pscol**

SYNTAXE [v :=] **pscol** *imin imout*

OBJET Quand l'image *imin* est à niveaux de gris, elle est coloriée avec la palette standard. Quand *imin* est binaire, chaque composante connexe de *imin* reçoit une couleur différente et le résultat est placé dans l'image *imout*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8*

VOIR AUSSI **clrcol, col, dscol**

PROCEDURE **cp**

SYNTAXE **cp** *imin imput*

OBJET Copie l'image *imin* avec sa palette dans l'image *imout*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8*

VOIR AUSSI **cppal**

PROCEDURE **cppal**

SYNTAXE **cppal** *imin imout*

OBJET Affecte la palette de l'image *imin* à l'image *imout*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8*

VOIR AUSSI **cp**

PROCEDURE refresh

SYNTAXE refresh imin

OBJET Visualise l'image *imin* au-dessus des autre images visualisées.

MODES grid : 1, 0 edge : 1, 0 depth : 1, 8, 16

PROCEDURE profile

SYNTAXE profile imin

OBJET Visualise le profil de gris d'une section verticale de l'image *imin*.

MODES grid : 1, 0 edge : 1, 0 depth : 8

PROCEDURE coldisplay

SYNTAXE coldisplay imin1 imin2 zoom "title"

OBJET Si *imin1* et *imin2* sont des images binaires, la procédure visualise la première en rouge et la seconde en vert. Si une seule image est binaire, elle est visualisée en rouge par-dessus l'image à niveaux de gris. "title" est le titre écrit dans la fenêtre de visualisation.

MODES grid : 1, 0 edge : 1, 0 depth : 1, 8

REMARQUE Une des deux images au moins doit être binaire. La visualisation des autres images est gelée tant que la procédure est active.

VOIR AUSSI dscol

3. EROSIONS, DILATATIONS

3.1 Erosions et Dilatations de base

PROCEDURE **dil**

SYNTAXE **dil** imin imout size

OBJET Dilatation binaire ou numérique (selon la profondeur de l'image d'entrée), de taille *size*, par un carré ou un hexagone (selon la trame choisie), en mode standard ou intrinsèque (selon la valeur de la clé "edge").

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

VOIR AUSSI **ero**

PROCEDURE **ero**

SYNTAXE **ero** imin imout size

OBJET Erosion binaire ou numérique (selon la profondeur de l'image d'entrée), de taille *size*, par un carré ou un hexagone (selon la trame choisie), en mode standard ou intrinsèque (selon la valeur de la clé "edge").

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

VOIR AUSSI **dil**

PROCEDURE **distance**

SYNTAXE **distance** binin greyout

OBJET Fonction distance, carrée ou hexagonale, à l'image *binin* (donc = 0 dans *binin*).

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1 to 8,16

REMARQUE Procédure moins isotrope, mais sensiblement plus rapide qu' **isodist**.

VOIR AUSSI **isodist**

PROCEDURE dirdil

SYNTAXE **dirdil** dir imin imout size

OBJET Dilatation binaire ou numérique (selon la profondeur de l'image d'entrée), par un segment de direction *dir* et de taille *size*, en trame carrée ou hexagonale (selon la valeur de la clé "grid"), en mode standard ou intrinsèque (selon la valeur de la clé "edge") . Les directions sont codées ainsi :

		6	1			8	1	2
hexagonal	5	0	2	;	carré	7	0	3
		4	3			6	5	4

où le symbole "0" indique l'origine de l'élément structurant .

MODES *grid* : 1 , 0 *edge* : 1 , 0 *depth* : 1 , 8 , 16

REMARQUE On notera que la procédure *décale* les images, puisque l'origine n' est pas au centre de l'élément structurant. Cela permet, en contrepartie, de prendre en compte les segments de longueur paire.

VOIR AUSSI **direro** , ainsi que **diropen** et **dirclose** dans OUVERTURES.

PROCEDURE direro

SYNTAXE **direro** dir imin imout size

OBJET Erosion binaire ou numérique (selon la profondeur de l'image d'entrée), par un segment de direction *dir* et de taille *size*, en trame carrée ou hexagonale (selon la valeur de la clé "grid"), en mode standard ou intrinsèque (selon la valeur de la clé "edge") . Les directions sont codées comme pour **dirdil**.

MODES *grid* : 1 , 0 *edge* : 1 , 0 *depth* : 1 , 8 , 16

REMARQUE Le commentaire sur le décalage, dans **dirdil**, s' applique aussi à **direro**

VOIR AUSSI **dirdil** , ainsi que **diropen** et **dirclose** dans OUVERTURES .

PROCEDURE **minidil**

SYNTAXE **minidil** imin imout

OBJET Dilatation élémentaire, binaire ou numérique (selon la profondeur de l'image d'entrée). Le mini élément structurant vaut trois points en trame hexagonale, et quatre en carré, disposés ainsi :

hexagonal : $\begin{matrix} & \underline{1} & \\ 1 & & 1 \end{matrix}$ carré : $\begin{matrix} & \underline{1} & 1 \\ 1 & & 1 \end{matrix}$

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

REMARQUE Noter le non-centrage, à comparer avec celui, dual pour l'ouverture, de **miniero**.

VOIR AUSSI **miniero**, ainsi que **miniopen** et **miniclose** dans OUVERTURES.

PROCEDURE **miniero**

SYNTAXE **miniero** imin imout

OBJET Erosion élémentaire, binaire ou numérique (selon la profondeur de l'image d'entrée). Le mini élément structurant vaut trois points en trame hexagonale, et quatre en carré, disposés ainsi :

hexagonal : $\begin{matrix} & & 1 \\ 1 & & 1 \\ & \underline{1} & \end{matrix}$ carré : $\begin{matrix} & & 1 \\ 1 & & 1 \\ & 1 & \underline{1} \end{matrix}$

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

REMARQUE Versions dédoublées : **binminiero** , **greyminiero** .Noter le non-centrage, à comparer avec celui, dual pour l'ouverture, de **minidil**.

VOIR AUSSI **minidil**, ainsi que **miniopen** et **miniclose** dans OUVERTURES.

PROCEDURE **b1dil**

SYNTAXE **b1dil** imin imout

OBJET Dilatation hexagonale de *imin* par *B1* , où :

$B1 = \begin{matrix} & & \bullet \\ \bullet & \bullet & \\ & & \bullet \end{matrix}$ $B2 = \begin{matrix} & \bullet & \\ \bullet & & \bullet \\ & \bullet & \end{matrix}$

MODES *grid* : 1 *edge* : 1 *depth* : 1, 8, 16

PROCEDURE **b2dil**

SYNTAXE **b2dil** imin imout

OBJET Dilatation hexagonale de *imin* par le *B2* défini en procédure **b1dil**

MODES *grid* : 1 *edge* : 1 *depth* : 1, 8, 16

PROCEDURE **dbldil**

SYNTAXE **dbldil** dir imin imout size

OBJET Dilatation binaire ou numérique (selon la profondeur de l'image d'entrée) , par un doublet de points de direction *dir* et distants de *size*, en mode standard . Les directions sont codées comme indiqué dans **dirdil** .

MODES *grid* : 1, 0 *edge* : 1 *depth* : 1, 8, 16

REMARQUE On notera que la procédure *décale* les images, puisque l'origine *n'* est pas au centre de l'élément structurant. Cela permet, en contrepartie, de prendre en compte les segments de longueur paire.

VOIR AUSSI **dirdil** , **dblero**

PROCEDURE **dblero** dir imin imout size

OBJET Erosion binaire ou numérique (selon la profondeur de l'image d'entrée) , par un doublet de points de direction *dir* et distants de *size* , en mode standard. Les directions sont codées comme indiqué dans **dirdil** .

MODES *grid* : 1, 0 *edge* : 1 *depth* : 1, 8, 16

REMARQUE Le commentaire sur le décalage, dans **dbldil**, s'applique aussi à **dblero** .

VOIR AUSSI **direro** , **dbldil**

PROCEDURE **gradient**

SYNTAXE **gradient** greyin greyout size

OBJET Gradient de Beucher, ou encore gradient morphologique, *i.e.* différence entre la dilatation **dil** et l'érosion **ero**, toutes deux de taille *size*.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 8, 16

PROCEDURE **sobel**

SYNTAXE **sobel** greyin greyout

OBJET Gradient de Sobel (opérateur de convolution).

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8, 16*

PROCEDURE **dilate**

SYNTAXE **dilate** imin imout size

OBJET Dilatation binaire ou numérique rapide (mais avec effets de bord), de taille *size*, par un carré ou un hexagone (selon la trame choisie).

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1, 8, 16*

VOIR AUSSI **erode**

PROCEDURE **erode**

SYNTAXE **erode** imin imout size

OBJET Erosion binaire ou numérique rapide avec effets de bord.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1, 8, 16*

VOIR AUSSI **dilate**

PROCEDURE **sq4cont**

SYNTAXE **sq4cont** binin binout

OBJET Contour interne, carré (4-connexité), de binin dans binout .

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1*

VOIR AUSSI **cont**

PROCEDURE **cont**

SYNTAXE **cont** binin binout

OBJET Contour interne, carré ou hexagonal, de binin dans binout. Le choix de *edge* n'est pas anodin.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 11*

3.2 Autres dilatations et érosions

PROCEDURE **isodil**

SYNTAXE **isodil** imin imout size

OBJET Dilatation isotrope, binaire ou numérique, en trame carrée ou hexagonale, de taille *size*. Procédure duale de **isoero** pour le complément.

MODES *grid* : 1, 0 *edge* : 1 *depth* : 1, 8, 16

PROCEDURE **isoero**

SYNTAXE **isoero** imin imout size

OBJET Erosion isotrope, binaire ou numérique, en trame carrée ou hexagonale, de taille *size*. Procédure duale de **isodil** pour le complément.

MODES *grid* : 1, 0 *edge* : 1 *depth* : 1, 8, 16

PROCEDURE **isodist**

SYNTAXE **isodist** binin greyout sizemax

OBJET Fonction "distance des points de binin à leur frontière" .

MODES *grid* : 1, 0 *edge* : 1 *depth* : 1

PROCEDURE **conedil**

SYNTAXE **conedil** binin greyout grey_display size type

OBJET Dilatation par le cône de base hexagonale (*type* = 0) ou dodécagonale (*type* = 1) de taille *size*, et de pente unité. L'origine est placée au milieu de la base pour *type* = 0, et au sommet du cône pour *type* = 1..

MODES *grid* : 1,[0] *edge* : 1, 0 *depth* : 1, to 8, 16

REMARQUE Tout fond sombre est relevé de la quantité (*unit_eight** *size*) . La procédure d'érosion duale n'est pas donnée, on la construit à partir de l'image greyn inversée.

VOIR AUSSI **cyldil** , **rhombodil**

PROCEDURE **cyldil****SYNTAXE** **cyldil** greyin greyout unit_eight size**OBJET** Dilatation par le cylindre de base hexagonale ou carrée (selon la trame), de taille *size*, et de hauteur (*unit_eight * size*) . L'origine est placée au milieu de la base.**MODES** *grid : 1 , 0* *edge : 1 , 0* *depth : 8 , 16***REMARQUE** Tout fond sombre est relevé de la quantité (*unit_eight* size*) . La procédure d'érosion duale n'est pas donnée, on peut la construire facilement à partir de l'image greyin inversée.**VOIR AUSSI** **conedil , rhombodil**

PROCEDURE **crossdil****SYNTAXE** **crossdil** binin1 binin2 binout**OBJET** Dilatation de l'ensemble *binin2* par le contour de l'ensemble *binin1*, donc par l'ensemble *binin1* lui-même. L'origine est prise au point en haut à gauche du champ (coordonnées : 0, 0).**MODES** *grid : [1], 0* *edge : 1 , 0* *depth : 1***REMARQUE** L'opération est d'autant plus rapide que *binin1* est plus proche de l'origine. Passage automatique en trame carrée durant la procédure.

PROCEDURE **rhombodil****SYNTAXE** **rhombodil** greyin greyout height size**OBJET** Dilatation par le rhombododécaèdre obtenu lui-même par dilatations des quatre demi diagonales du prisme droit de base carrée (*size*size*), et de hauteur (*height*size*), centré à l'origine.**MODES** *grid : [1], 0* *edge : 1* *depth : 8 , 16***REMARQUE** Passage automatique en trame carrée pendant la procédure. Décalage de l'image, vers le haut, de la valeur *height*.**VOIR AUSSI** **conedil, cyldil**

PROCEDURE **rhomboero**

SYNTAXE **rhomboero** greyin greyout height size

OBJET Erosion par le rhombododécaèdre obtenu lui-même par dilatations des quatre demi diagonales du prisme droit de base carrée ($size*size$), et de hauteur ($height*size$), centré à l'origine.

MODES $grid : [1], 0$ $edge : 1, 0$ $depth : 8, 16$

REMARQUE Les mêmes que pour **rhombodil**, plus la possibilité de travailler en modes standard ou intrinsèque.

VOIR AUSSI **rhombodil**

PROCEDURE **bero**

SYNTAXE **bero** imin imout

OBJET Procédure, duale pour le complément, du produit (**b1dil** o **b2dil**)

MODES $grid : 1$ $edge : 1$ $depth : 1, 8, 16$

PROCEDURE **diamdil**

SYNTAXE **diamdil** imin imout size

OBJET Produit de $size$ dilatations de imin par le diamant D :

$$D = \begin{array}{ccc} & \bullet & \\ \bullet & \bullet & \bullet \\ & \bullet & \end{array}$$

MODES $grid : 0$ $edge : 1$ $depth : 1$

VOIR AUSSI **diamero**

PROCEDURE **diamero**

SYNTAXE **diamero** imin imout size

OBJET Produit de $size$ érosions de imin par le diamant D de **diamdil**

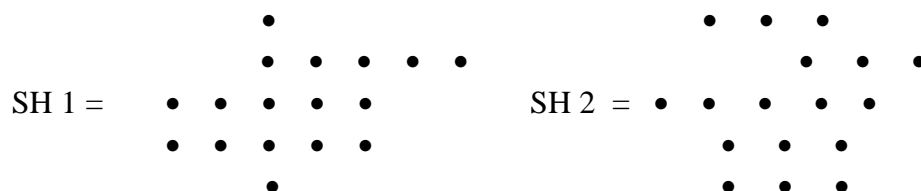
MODES $grid : 0$ $edge : 1$ $depth : 1, 8, 16$

REMARQUE Procédure duale de **bdiamdil** pour le complément.

PROCEDURE sh1dil

SYNTAXE sh1dil imin imout

OBJET Dilatation de *imin* par l'hexagone SH 1:



MODES grid : 0 edge : 1 depth : 1, 8, 16

PROCEDURE sh2dil

SYNTAXE sh2dil imin imout

OBJET Dilatation de *imin* par l'hexagone SH 2 représenté en procédure **sh1dil**.

MODES grid : 0 edge : 1 depth : 1, 8, 16

PROCEDURE sh1ero

SYNTAXE sh1ero binin binout

OBJET Erosion duale de **sh1dil** pour le complément.

MODES grid : 0 edge : 1 depth : 1

VOIR AUSSI sh1dil

PROCEDURE sh2ero

SYNTAXE sh2ero imin imout

OBJET Erosion duale de **sh2dil** pour le complément.

MODES grid : 0 edge : 1 depth : 1, 8, 16

VOIR AUSSI sh2dil

PROCEDURE **shdil**

SYNTAXE **shdil** *imin imout size*

OBJET Produit de composition (**sh1dil** o **sh2dil**) répété *size* fois.

MODES *grid : 0* *edge : 1* *depth : 1, 8, 16*

REMARQUE L'élément structurant de cette procédure est un dodécagone régulier construit en trame carrée.

VOIR AUSSI **shero**

PROCEDURE **shero**

SYNTAXE **shero** *imin imout size*

OBJET Procédure duale de **shdil** pour le complément .

MODES *grid : 0* *edge : 1* *depth : 1, 8, 16*

VOIR AUSSI **shdil**

PROCEDURE **ringdil1**

SYNTAXE **ringdil1** *imin imout size*

SYNTAXE Dilatation de *imin* par les six sommets d'un hexagone de taille *size*. Le résultat est placé dans *imout*.

MODES *grid : 0* *edge : 1* *depth : 1, 8*

VOIR AUSSI **ringdil2**

PROCEDURE **ringdil2**

SYNTAXE **ringdil2** *imin imout size*

OBJET Dilatation de *imin* par le contour d'un hexagone de taille *size*. Le résultat est placé dans *imout*.

MODES *grid : 0* *edge : 1* *depth : 1, 8*

VOIR AUSSI **ringdil1**

4. OPERATEURS DE RANG

PROCEDURE **hexrank**

SYNTAXE **hexrank** imin imout rank_order

OBJET Opérateur de rang, pour images binaires ou numériques (selon la profondeur de l'image *imin*), en trame hexagonale. La taille du voisinage vaut 1 (fenêtre de 7 points) et l'ordre du rang varie de 1 à 7 .

MODES *grid* : 1 *edge* : 1 , [0] *depth* : 1 , 8 , 16

REMARQUE Le rang d' ordre 1 n' est autre que la dilatation, celui d' ordre 4 la médiane, et celui d' ordre 7 l'érosion. Les rangs 2 et 3 sont duaux des 5 et 6. La procédure se décline dans les trois versions **rank2**, **rank3**, **rank4**. L'opération est toujours traitée en mode standard.

VOIR AUSSI **median**

PROCEDURE **median**

SYNTAXE **median** imin imout

OBJET Médiane, pour images binaires ou numériques (selon la profondeur de l'image *imin*), en trame hexagonale ou carrée (selon la clé *grid*). Le voisinage est l'hexagone unitaire de 7 points en trame hexagonale, et le diamant élémentaire de 5 points en trame carrée. L'opération est toujours traitée en mode standard.

MODES *grid* : 1 , 0 *edge* : 1 , [0] *depth* : 1 , 8 , 16

REMARQUE **hexrank**

PROCEDURE **binsegmi**

SYNTAXE **binsegmi** binin binout size

OBJET Intersection de dilatations de l'ensemble *binin* par trois segments égaux, en hexagonal, ou deux en trame carrée. Les segments sont orientés selon les directions principales de la trame, et ont *size* pour longueur.

MODES *grid* : 1 , 0 *edge* : 1 , 0 *depth* : 1

5. CONVOLUTIONS

PROCEDURE **hgauss1**

SYNTAXE **hgauss1** greyin greyout

OBJET Convolution gaussienne de taille 1 dans la direction horizontale. Elle a pour support trois points, de poids successifs 1/4 ; 1/2 ; 1/4 .

MODES *grid : 0* *edge : 1* *depth : 8, 16*

REMARQUE Léger biais assombrissant, dû aux divisions entières de Micromorph.

VOIR AUSSI **vgauss1**

PROCEDURE **vgauss1**

SYNTAXE **vgauss1** greyin greyout

OBJET Convolution gaussienne de taille 1 dans la direction verticale. Elle a pour support trois points successifs, de poids 1/4 ; 1/2 ; 1/4.

MODES *grid : 0* *edge : 1* *depth : 8, 16*

REMARQUE Léger biais assombrissant, dû aux divisions entières de Micromorph.

VOIR AUSSI **hgauss1**

PROCEDURE **gauss1**

SYNTAXE **gauss1** greyin greyout

OBJET Convolution gaussienne isotrope de taille 1, qui s'exprime en trame carrée par les poids suivants :

1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16

MODES *grid : 0* *edge : 1* *depth : 8, 16*

REMARQUE La procédure **gauss1** s'obtient par produit de composition des deux procédures **hgauss1** et **vgauss1**.

PROCEDURE **gauss**

SYNTAXE **gauss** greyin greyout size

OBJET Convolution de taille égale à *size*. Elle s'obtient par itérations de la procédure **gauss1**.

MODES *grid: 0* *edge: 1* *depth: 8,16* .

6. OUVERTURES, FERMETURES

6.1 Ouvertures, fermetures de base

PROCEDURE **open**

SYNTAXE **open** imin imout size

OBJET Ouverture binaire ou numérique (selon la profondeur de l'image d'entrée), de taille *size* , par un carré ou un hexagone (selon la trame choisie), en mode standard .

MODES *grid* : 1 , 0 *edge* : 1 *depth* : 1, 8, 16

REMARQUE La procédure est produit de composition de **ero** par **dil**.

VOIR AUSSI **close**

PROCEDURE **close**

SYNTAXE **close** imin imout size

OBJET Fermeture binaire ou numérique (selon l'image d'entrée), de taille *size* , par un carré ou un hexagone (selon la trame choisie), en mode standard .

MODES *grid* : 1 , 0 *edge* : 1 *depth* : 1, 8, 16

REMARQUE La procédure est la duale de **open** pour le complément.

VOIR AUSSI **open**

PROCEDURE **diopen**

SYNTAXE **diopen** dir imin imout size

OBJET Ouverture binaire ou numérique (selon la profondeur de l'image d'entrée), par un segment de direction *dir* et de taille *size*, en trame carrée ou hexagonale (selon la valeur de la clé "grid"), en mode standard. Les directions sont codées ainsi :

		6	1			8	1	2
hexagonal	5	0	2	;	carré	7	0	3
		4	3			6	5	4

où le symbole "0" indique l'origine de l'élément structurant .

MODES *grid : 1 , 0* *edge : 1* *depth : 1, 8, 16*

REMARQUE La procédure **diropen** est le produit de composition de **direro** par **dirdil** .

VOIR AUSSI **dirclose**

PROCEDURE **dirclose**

SYNTAXE **dirclose** dir imin imout size

OBJET Fermeture duale de **diropen** pour le complément.

MODES *grid : 1 , 0* *edge : 1* *depth : 1, 8, 16*

REMARQUE On obtient **dirclose** par composition de **dirdil** de **direro**.

VOIR AUSSI **diropen**

PROCEDURE **lineopen**

SYNTAXE **lineopen** imin imout size

OBJET Sup, dans le cas numérique, ou réunion, dans le cas binaire, des opérations diropen de taille *size*, et selon les trois (resp. les deux) directions principales de la trame hexagonale (resp. carrée) .

MODES *grid : 1 , 0* *edge : 1* *depth : 1, 8, 16*

VOIR AUSSI **lineclose**

PROCEDURE **lineclose**

SYNTAXE **lineclose** imin imout size

MODES *grid : 1 , 0* *edge : 1* *depth : 1, 8, 16*

OBJET Opération duale de **lineopen** pour le complément.

PROCEDURE **openth**

SYNTAXE **openth** greyin greyout size

OBJET Résidu de l'ouverture **open**, carrée ou hexagonale, de taille *size* (extrait donc les pics et les lignes de crête relativement clairs). Cette opération est appelée *chapeau haut de forme par ouverture de disques* .

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8, 16*

PROCEDURE **closeth**

SYNTAXE **closeth** greyin greyout size

OBJET Résidu de la fermeture **close**, carrée ou hexagonale, de taille *size* (extrait donc les creux et les lignes de thalweg relativement sombres). Cette opération est appelée *chapeau haut de forme par fermeture de disques*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8, 16*

PROCEDURE **lineopenth**

SYNTAXE **lineopenth** greyin greyout size

OBJET Résidu de l'ouverture **lineopen**, en trame carrée ou hexagonale, de taille *size* (extrait donc les pics et les lignes de crête relativement clairs). Cette opération est appelée *chapeau haut de forme par union d'ouvertures linéaires*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8, 16*

PROCEDURE **linecloseth**

SYNTAXE **linecloseth** greyin greyout size

OBJET Résidu de la fermeture **lineclose**, en trame carrée ou hexagonale, de taille *size* (extrait donc les creux et les lignes de thalweg relativement sombres). Cette opération est appelée *chapeau haut de forme par intersection de fermetures linéaires*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8, 16*

PROCEDURE **pregrad**

SYNTAXE **pregrad** imin imout size

OBJET Gradient de Beucher régularisé de l'image de gris *imin*, placé dans l'image de gris *imout*. Le paramètre de taille *size* va de 1 à 10.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1, 8*

VOIR AUSSI **gradient, regrad**

PROCEDURE **regrad**

SYNTAXE **regrad** *imin imout*

OBJET Supremum de gradients de Beucher régularisés de l'image *imin*, placé dans l'image *imout*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1, 8*

VOIR AUSSI **gradient, pregrad**

6.2 Autres ouvertures et fermetures

PROCEDURE **miniopen**

SYNTAXE **miniopen** imin imout

OBJET Ouverture binaire ou numérique (selon la profondeur de l'image d'entrée), en mode standard, par le triangle élémentaire pointe en *bas* (trame hexagonale) ou par le carré de 4 pixels (trame carrée)

MODES *grid : 1, 0* *edge : 1* *depth : 1, 8, 16*

REMARQUE Peut servir de premier pas dans une granulométrie construite à partir de la procédure **open**.

PROCEDURE **miniclose**

SYNTAXE **miniclose** imin imout .

OBJET Opération duale de miniopen pour le complément, donc avec le même élément structurant (et non pas son transposé).

MODES *grid : 1, 0* *edge : 1* *depth : 1, 8, 16*

PROCEDURE **isopen**

SYNTAXE **isopen** imin imout taille

OBJET Ouverture isotrope binaire ou numérique (selon la profondeur de l'image d'entrée), en mode standard.

MODES *grid : 1, 0* *edge : 1* *depth : 1, 8, 16*

REMARQUE Cette procédure utilise **isodil** et **isoero**.

PROCEDURE **isoclose**

SYNTAXE **isoclose** imin imout taille.

OBJET Opération duale de isopen.

MODES *grid : 1, 0* *edge : 1* *depth : 1, 8, 16*

PROCEDURE **infopen**

SYNTAXE **infopen** imin imout accuracy size.

OBJET Ouverture limite, obtenue par itération de l'intersection, en binaire, ou de l'inf, en numérique, des trois (en hexagonal), ou des deux (en carré), ouvertures **diropen** de taille *size*, et selon les directions principales de la trame.

MODES *grid : 1, 0* *edge : 1* *depth : 1, 8, 16*

REMARQUE "accuracy" est un entier positif ou nul qui indique à partir de quand la différence des surfaces (en binaire), ou des volumes (en numérique), entre deux itérations successives peut être considérée comme négligeable. Lorsqu'il vaut zéro, on obtient la limite idempotente *sensu stricto*.

VOIR AUSSI **supclose**

PROCEDURE **supclose**

SYNTAXE **supclose** imin inout accuracy size.

OBJET Procédure duale de **infopen** pour le complément (il s'agit donc d'unions, ou de sup, de fermetures **dirclose**) .

MODES *grid : 1, 0* *edge : 1* *depth : 1, 8, 16*

VOIR AUSSI **infopen**

PROCEDURE **stopen**

SYNTAXE **stopen** imin imout size

OBJET Etape élémentaire de **infopen**

MODES *grid : 1, 0* *edge : 1* *depth : 1, 8, 16*

VOIR AUSSI **infopen**

7. GEODESIE ET CONNEXITE

Il est à noter que beaucoup d'opérations présentées dans cette section dérivent du mot primitif **imbuildngb**.

PROCEDURE **gdsdil**

SYNTAXE **gdsdil** imin mask imout size

OBJET Dilatation géodésique de taille *size* de *imin* dans le masque *mask*. Le résultat est placé dans *imout*. L'image à dilater, ainsi que le masque, peuvent être binaire ou numérique.

MODES *grid* : 1 , 0 *edge* : 1 *depth* : 1, 8, 16.

PROCEDURE **gdsero**

SYNTAXE **gdsero** binin mask imout size

OBJET Erosion duale de **gdsdil** pour le complément.

MODES *grid* : 1 , 0 *edge* : 1 *depth* : 1, 8, 16

PROCEDURE **build**

SYNTAXE **build** mask iminout

OBJET Dilatation géodésique infinie de l'image *iminout* dans l'image *mask*. Le résultat est placé dans la mémoire de l'image à dilater. Considérée comme une opération sur *iminout*, à *mask* fixé, **build** est une fermeture; considérée au contraire comme une transformation de *mask*, à *iminout* fixée, il s'agit de *l'ouverture connexe* de *mask* marqué par *iminout*.

MODES *grid* : 1 , 0 *edge* : 1 *depth* : 1, 8, 16

PROCEDURE **buildopen**

SYNTAXE **buildopen** imin imout size

OBJET Ouverture par reconstruction d'*imin* dans *imout*, de taille *size*. Cette procédure retrouve la procédure **build**, où *mask* est remplacé par *imin* et où *iminout* est remplacé par l'érodé **ero** de taille *size* de *imin*. Toutefois, le résultat est placé ici dans la mémoire *imout*, éventuellement différente de *imin*.

MODES *grid* : 1 , 0 *edge* : 1 *depth* : 1, 8, 16

PROCEDURE **buildclose**

SYNTAXE **buildclose** imin imout size

OBJET Fermeture par reconstruction d'*imin* dans *imout*, de taille *size*.

MODES *grid* : 1 , 0 *edge* : 1 *depth* : 1, 8, 16

PROCEDURE **areaopen**

SYNTAXE **areaopen** binin binout size

OBJET Place en *binout* toutes les particules d'aire supérieure ou égale à *size*, mesurées en nombre de pixels.

MODES *grid* : 1 , 0 *edge* : 1 *depth* : 1

PROCEDURE **ringbuild**

SYNTAXE **ringbuild** imin immask imout size

OBJET Reconstitue *imin* à partir du masque *immask*, au moyen de dilatations par le contour d'un hexagone de taille *size*.

MODES *grid* : 1 , 0 *edge* : 1 *depth* : 1, 8

VOIR AUSSI **build, buildopen, buildclose**

PROCEDURE **gdsdist**

SYNTAXE **gdsdist** binin binmask greyout

OBJET Fonction distance géodésique de l'image binaire *binin*, à l'intérieur du masque binaire *binmask*. Le résultat est placé dans l'image numérique *greyout*.

MODES *grid* : 1 , 0 *edge* : 1 *depth* : 1, 8

VOIR AUSSI **distance**

PROCEDURE **recons**

SYNTAXE **recons** imin mask imout type

OBJET Reconstruction numérique de *imin* à l'intérieur de *mask*, le résultat est placé dans *imout*. Si *type=0*, la reconstruction travaille sur les images d'entrée et conduit à l'ouverture par reconstruction. Si *type=1*, la procédure travaille de façon duale et conduit à la fermeture par reconstruction.

MODES *grid* : 1 , 0 *edge* : 1 *depth* : 8, 16

VOIR AUSSI **build**

PROCEDURE **levelling**

SYNTAXE **levelling** imin immask imout

OBJET Place dans *imout* le nivellement de *imin* selon le marqueur *immask*.

MODES *grid* : 1 , 0 *edge* : 1 *depth* : 1, 8, 16

REMARQUE Le nivellement est le produit commutatif d'une ouverture par reconstruction, pour un marqueur donné, par la fermeture par reconstruction, pour le même marqueur.

VOIR AUSSI **recons, buildopen, buildclose**

8. APPLICATIONS DE LA GEODESIE

PROCEDURE **edgeoff**

SYNTAXE **edgeoff** binin binout

OBJET Supprime les composantes connexes touchant les bords du champ.

MODES *grid* : 1, 0 *edge* : 1 *depth* : 1

PROCEDURE **fgrain**

SYNTAXE **fgrain** binin binout

OBJET Place dans binout la première composante connexe de *binin*, rencontrée dans le sens du balayage.

MODES *grid* : 1, 0 *edge* : 1 *depth* : 1

REMARQUE La procédure modifie *binin* en lui supprimant sa première composante connexe. Cette circonstance, exceptionnelle, se justifie par le fait qu'elle permet de placer **fgrain** à l'intérieur de boucles de traitements.

VOIR AUSSI **areaopen**

PROCEDURE **border**

SYNTAXE **border** binout

OBJET Place en *binout* le contour du bord du champ.

MODES *grid* : 1, 0 *edge* : [1], 0 *depth* : 1, 8, 16

PROCEDURE **clohole**

SYNTAXE **clohole** imin imout

OBJET Bouche les pores internes de *imin* (ou comble les creux si l'on est en numérique), et place l'image transformée dans *imout*.

MODES *grid* : 1, 0 *edge* : 1 *depth* : 1, 8, 16

PROCEDURE **maxima**

SYNTAXE **maxima** greyin binout

OBJET Place en *binout* les maxima de *greyin* .

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8, 16 to 1*

VARIANTE **minima**

PROCEDURE **extmaxima**

SYNTAXE **extmaxima** greyin binout height

OBJET Place en *binout* les maxima de *greyin* qui sont entourés d'une zone non ascendante de hauteur au moins égale à *height*

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8, 16 to 1*

PROCEDURE **extminima**

SYNTAXE **extminima** greyin binout height

OBJET Place en *binout* les minima de *greyin* qui sont entourés d'une zone non descendante de hauteur au moins égale à *height*

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8, 16 to 1*

PROCEDURE **swamping**

SYNTAXE **swamping** greyin binmark greyout

OBJET Modification de l'homotopie de *greyin* pour lui imposer toutes les composantes connexes de *binmark* comme minima, et elles seulement. On procède par fermeture par reconstruction de *greyin* relativement au complément de *binmark*, considéré comme fonction numérique.

MODES *grid : 1, 0* *edge : 1* *depth : 8, 16*

PROCEDURE **extrema**

SYNTAXE **extrema** imin immask binout1 binout2

OBJET Fournit dans *binout1* (resp. *binout2*) l'ensemble des valeurs les plus hautes (resp. les plus basses) de *greyin* à l'intérieur de *binmask*. Les valeurs extrêmes sont imprimées.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8

VOIR AUSSI **maxima, minima, extmaxima, extminima**

PROCEDURE **grainclose**

SYNTAXE **grainclose** imin imout size1 size2

OBJET Supprime les petits objets de *imin* par une ouverture par reconstruction de taille 1. L'image résultat est alors fermée par une fermeture de taille 2 et on effectue le supremum avec l'image initiale (afin de récupérer les petits objets). Le résultat est placé dans *imout*.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

REMARQUE Par itération, **grainclose** devient une fermeture. En pratique, on atteint l'idempotence quasiment après la première étape. En prenant *size1=0*, seule l'étape de fermeture est effectuée.

VOIR AUSSI **indivaf**

PROCEDURE **indivaf**

SYNTAXE **indivaf** binin binout size1 size2

OBJET La procédure effectue d'abord une ouverture par reconstruction de taille *size1* de l'image binaire *binin*, puis une fermeture individuelle de chaque particule avant de prendre l'union des résultats.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1

REMARQUE Au départ, *indivaf* ressemble à un filtre alterné séquentiel. En fait, l'idempotence est pratiquement atteinte après la première étape. En prenant *size1=0*, on supprime la première phase d'ouverture par reconstruction.

VOIR AUSSI **grainclose**

PROCEDURE **dynamics**

SYNTAXE **dynamics** *imin imout height type*

OBJET Sélection des maxima/minima de l'image *imin* dont la dynamique est supérieure à *height*. Si *type=0*, les minima sont extraits, sinon ce sont les maxima. Dans les deux cas, ils sont placés dans l'image de gris *imout*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8*

VOIR AUSSI **extmaxima, extminima**

9. FILTRES MORPHOLOGIQUES

9.1 Filtres principaux

Dans la syntaxe de chacun des filtres de cette section, le paramètre d'état binaire "type" permet de commencer l'opération soit par une fermeture ($t=1$), soit par une ouverture ($t\neq 1$).

PROCEDURE **af**

SYNTAXE **af** imin imout size type

OBJET Filtre alterné de base; *i.e.* produit de composition de **close** par **open**, si l'on a $type = 1$, ou de **open** par **close**, si $type \neq 1$.

MODES *grid* : 1, 0 *edge* : 1 *depth* : 1, 8, 16

PROCEDURE **fullasf**

SYNTAXE **fullasf** imin imout size type

OBJET Filtre alterné séquentiel par pas de 1 construit sur **af**. En d' autres termes:

$\text{fullasf}(size) := \text{af}(size) \circ \dots \circ \text{af}(3) \circ \text{af}(2) \circ \text{af}(1)$.

MODES *grid* : 1, 0 *edge* : 1 *depth* : 1, 8, 16

REMARQUE Le paramètre type conserve la même signification que dans **af**.

VOIR AUSSI **asf**

PROCEDURE **lineaf**

SYNTAXE **lineaf** imin imout size type

OBJET Filtre alterné construit sur **lineopen** et **lineclose**. L' ordre est déterminé par *type* comme dans **af** ($type = 1 \Rightarrow$ fermeture d' abord).

MODES *grid* : 1, 0 *edge* : 1 *depth* : 1, 8, 16

PROCEDURE **asf**

SYNTAXE **asf** imin imout size type

OBJET Filtre alterné séquentiel par pas croissants construit sur **af**. La progression admet pour loi :

$$size(i) = size(i - 1) + i - 1,$$

ce qui donne, pour les premiers pas, la série 1, 2, 4, 7, 11, 16....

On a donc:

$$\mathbf{asf}(10) = \mathbf{asf}(7) = \mathbf{af}(7) \circ \mathbf{af}(4) \circ \mathbf{af}(2) \circ \mathbf{af}(1).$$

MODES *grid* : 1, 0 *edge* : 1 *depth* : 1, 8, 16

REMARQUE Procédure sensiblement plus rapide que **fullasf**, aux grandes valeurs surtout, et à peu près aussi fine .

VOIR AUSSI **fullasf**

PROCEDURE **lineasf**

SYNTAXE **lineasf** imin imout size type

OBJET Filtre alterné séquentiel construit sur **lineaf** selon la même progression des tailles que pour **asf**.

MODES *grid* : 1, 0 *edge* : 1 *depth* : 1, 8, 16

PROCEDURE **automed**

SYNTAXE **automed** imin imout size

OBJET Centre morphologique entre les deux filtres alternés **af** (*size*) de type 1 et 0.

MODES *grid* : 1, 0 *edge* : 1 *depth* : 1, 8, 16

PROCEDURE **centre**

SYNTAXE **centre** imin imin1 imin2 imout

OBJET Centre morphologique entre les deux images *imin1* et *imin2*, relativement à l'image de référence *imin*

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16.

PROCEDURE **contrast**

SYNTAXE **contrast** greyin greyout sz type

OBJET Contraste $\kappa(f)$ sur *greyin* := f, défini comme suit :

Si $\xi(f)(x) - f(x) \leq f(x) - \eta(f)(x)$ alors $\kappa(f)(x) = \xi(f)(x)$
sinon $\kappa(f)(x) = \eta(f)(x)$
avec type = 1 $\Rightarrow \xi(f) := \mathbf{isodil}(f;sz)$ $\eta(f) := \mathbf{isoero}(f;sz)$
type = 2 $\Rightarrow \xi(f) := \mathbf{isopen}(f;sz)$ $\eta(f) := \mathbf{isoclose}(f;sz)$
type > 2 $\Rightarrow \xi(f) := \mathbf{isopen}(f;sz)$ $\eta(f) := \mathbf{isoclose}(f;5 \times sz)$.

MODES *grid* : 1, 0 *edge* : 1 *depth* : 8, 16

PROCEDURE **contrasth**

SYNTAXE **contrasth** greyin greyout sz

OBJET Contraste $\kappa(f)$ sur *greyin*:=f, par chapeaux haut de forme clair et sombre i.e.:

$\kappa(f) := 3f - \mathbf{open}(f;sz) - \mathbf{close}(f;sz)$

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 8, 16

9.2 Filtres supplémentaires

PROCEDURE **minifilt**

SYNTAXE **minifilt** imin imout

OBJET Filtre alterné obtenu par composition de **miniopen** et de **miniclose** .

MODES *grid* : 1 , 0 *edge* : 1 *depth* : 1 , 8 , 16

PROCEDURE **isaf**

SYNTAXE **isaf** imin imout size type

OBJET Filtre alterné isotrope de taille *size* , en binaire ou en numérique, en carré ou en hexagonal. Le paramètre d'état binaire *type* permet de commencer l'opération soit par une fermeture (*type* = 1), soit par une ouverture (*type* ≠ 1).

MODES *grid* : 1 , 0 *edge* : 1 *depth* : 1 , 8 , 16

VOIR AUSSI **isaf**

PROCEDURE **isaf**

SYNTAXE **isaf** imin imout size type

OBJET Filtre alterné séquentiel isotrope de taille *size*, en binaire ou en numérique, en carré ou en hexagonal. Le paramètre d'état binaire *type* permet de commencer l'opération soit par une fermeture (*type* = 1), soit par une ouverture (*type* ≠ 1).

MODES *grid* : 1 , 0 *edge* : 1 *depth* : 1 , 8 , 16.

PROCEDURE **buildaf**

SYNTAXE **buildaf** imin imout size type

OBJET filtre alterné par reconstruction; *i.e.* produit de composition de **buildopen** par **buildclose** si *type*=0, ou de **buildclose** par **buildopen** si *type* = 1 .

MODES *grid* : 1 , 0 *edge* : 1 *depth* : 1 , 8 , 16

VOIR AUSSI **buildaf**

PROCEDURE **buildasf**

SYNTAXE **buildasf** imin imout size type

OBJET Filtre alterné séquentiel par pas croissants construit sur **buildaf**. La progression admet pour loi :

$$size(i) = size(i-1) + i - 1,$$

ce qui donne, pour les premiers pas, la série 1, 2, 4, 7, 11, 16....On a donc:

buildasf (10)=**buildasf**(7)=**buildaf**(7)ο**buildaf**(4)ο**buildaf**(2)ο **buildaf**(1) .

MODES *grid* : 1 , 0 *edge* : 1 *depth* : 1 , 8 , 16

REMARQUE Le paramètre *type* conserve la même signification que dans **buildaf** .

PROCEDURE **minibuildaf**

SYNTAXE **minibuildaf** imin imout

OBJET Produit de l'ouverture par reconstruction à l'aide de **miniopen** par la fermeture par reconstruction à l'aide de **miniclose**.

MODES *grid* : 1,0 *edge* : 1,0 *depth* : 1,8,16

VOIR AUSSI **miniopen**, **miniclose**

PROCEDURE **closoropen**

SYNTAXE **closoropen** greyin greyout size

OBJET Toggle entre **gbuildopen**, **gbuildclose**, de taille *size*, et l'identité. Si, au point x, l'identité vaut la fermeture, alors on prend l'ouverture ; si elle vaut l'ouverture, alors on prend la fermeture ; sinon on reste inchangé.

MODES *grid* : 1, 0 *edge* : 1 , 0 *depth* : 8, 16

PROCEDURE **binmiddle**

SYNTAXE **binmiddle** binin1 binin2 binout

OBJET Construit l'ensemble médian entre *binin1* et *binin2* et le place dans *binout*.

MODES *grid* : 1, 0 *edge* : 1 , 0 *depth* : 1

REMARQUE L'intersection entre *binin1* et *binin2* doit être non vide.

VOIR AUSSI **grmiddle**

PROCEDURE **grimiddle**

SYNTAXE **grmiddle** *greyin1* *greyin2* *greyout*

OBJET Génère l'image à niveaux de gris médiane entre *greyin1* et *greyin2* et place le résultat dans *greyout*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8*

VOIR AUSSI **binmiddle**

10. BOULES MAXIMALES ET SQUELETTE

PROCEDURE **binopenskel**

SYNTAXE **binopenskel** binin binout

OBJET Squelette de *binin*, au sens des carrés (ou des hexagones) maximaux inscriptibles, placé en *binout*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1*

PROCEDURE **binultim**

SYNTAXE **binultim** binin binout

OBJET Erodés ultimes de *binin*, placés dans *binout*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1*

VOIR AUSSI **centroid**

PROCEDURE **centroid**

SYNTAXE **centroid** binin binout

OBJET Dans binout, **fpoint** de l'ultime des érodés de *binin*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1*

REMARQUE le centroïde est toujours inclus dans l'ensemble de départ.

PROCEDURE **condbis**

SYNTAXE **condbis** binin binout size

OBJET Bissectrice conditionnelle de l'image binaire *binin*, le résultat est placé dans l'image binaire *binout*. Le paramètre *size* s'étend de 1 à 10.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1*

11. AMINCISSEMENTS, EPAISSISSEMENTS

11.1 Amincissements, épaisissements binaires

Les amincissements et épaisissements présentés ici sont tous construits en trois fois . Il vient d'abord l'étape élémentaire. Par exemple, pour l'épaissement de X par l'élément structurant $\{se1(\alpha), se0(\alpha)\}$, cette étape s'écrit :

$$Y(X; se1(a), se0(a)) = Y(X, a) = X \cup (X \ominus se1) \cap (X^c \ominus se0) \quad (1)$$

puis vient le produit de composition des $Y(X,\alpha)$ lorsque l'élément structurant subit les six ou huit rotations de la trame, *i.e.* en hexagonal, par exemple:

$$S(X) := Y(X, \alpha+5\pi/6) \circ Y(X, \alpha+4\pi/6) \circ \dots \circ Y(X, \alpha) \quad (2)$$

et enfin l'itération du cycle $S(X)$ jusqu'à idempotence. Dans cette section, Seules les deux dernières étapes font l'objet de procédures (la première peut s'obtenir directement à partir du mot primitif **imhitormiss**). Ainsi, la relation (2) est donnée par la procédure **thickturn**, et son itérée limite par **thick**. Dans le cas d'amincissement, $Y(X,\alpha)$ est remplacé par:

$$Y'(X, a) := X / (X \ominus se1) \cap (X^c \ominus se0).$$

On notera que la dualité pour le complément *intervertit* les deux parties de l'élément structurant. L'étape élémentaire devient :

$$Y'(X ; se1, se0) = [Y(X^c; se0, se1)]^c \quad (3)$$

et l'interversion s'étend aux couples **thickturn-thinturn**, puis **thick-thin**. En ce qui concerne les modes, l'une ou l'autre des représentations standard ou transitive peut être avantageuse selon l'élément structurant choisi. Par conséquent, *pour la présente section*, la clé $edge=1$ correspond au cas standard (comme d'habitude), mais $edge = 0$ au cas transitif (et non intrinsèque comme dans les chapitres d' opérateurs croissants). Sont particularisées les trois procédures **thick** (et **thin**) associées à des opérateurs homotopiques en trame hexagonale, complétés par leurs homologues en trame carrée.

La construction en trois étages se retrouve aussi dans les versions géodésiques des algorithmes. La seule différence se situe au niveau de l'étape élémentaire (1), qui devient :

$$Y(X, a) := [X \cup (X \ominus se1) \cap (X^c \ominus se0)] \cap \text{mask}$$

où "mask" désigne le masque géodésique. C' est pourquoi on se contentera de présenter les versions géodésiques comme des variantes de procédures. A noter aussi que ces versions n'ont de sens, par définition, qu'en mode standard.

PROCEDURE **thickturn**

SYNTAXE **thickturn** *se1 se0 binin binout*

OBJET Placé en *binout*, le résultat *S* du cycle des épaisissements élémentaires de *binin* selon le doublet $\{se1(\alpha); se0(\alpha)\}$ et ses rotations. Si $binin = X$, on a donc $binout = S$, au sens de la relation (2) .

MODES *grid : 1, 0* *edge : 1, [0]* *depth : 1*

VARIANTE Version géodésique, en mode standard, de syntaxe

gdsthickturn *se1 se0 binin mask binout*

VOIR AUSSI **thinturn, gdsthinturn**

PROCEDURE **thick**

SYNTAXE **thick** *se1 se0 binin binout*

OBJET Limite (idempotente) des itérés de "**thickturn** *se1 se0 binin binout* "

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1*

VARIANTE Epaisseur géodésique **gdsthick**, en mode standard, de syntaxe

gdsthick *se1 se0 binin mask binout*

VOIR AUSSI **thin, gdsthin**

PROCEDURE **thinturn**

SYNTAXE **thinturn** *se1 se0 binin binout*

OBJET Amincissement dual de **thickturn** pour la complémentation, (c.f. rel. 3)

MODES *grid : 1, 0* *edge : [1], 0* *depth : 1*

REMARQUE Noter l'interversion des deux éléments structurants *se1* et *se0* dans la relation de dualité (3) .

VARIANTE Version géodésique **gdsthinturn**, obtenue par application de la relation (3) à **gdsthickturn**, dont le mode est uniquement (et automatiquement) standard .

PROCEDURE thin

SYNTAXE thin binin binout

OBJET Limite (idempotente) des itérés de "**thinturn** se1 se0 binin binout"

MODES grid : 1 , 0 edge : 1 , 0 depth : 1

VARIANTE Epaisseur géodésique "**gdsthin** se1 se0 binin mask binout", obtenu par itérations de **gdsthinturn** jusqu'à idempotence.

VOIR AUSSI thick et gdsthick

PROCEDURE Lthick

SYNTAXE Lthick binin binout

OBJET Particularisation de l'épaissement **thick** à l'élément structurant L, i.e.:

		0	0			0	0	0
hexagonal	*	*	*	;	carré	*	*	*
		1	1			1	1	1

MODES grid : 1 , 0 edge : 1 , 0 depth : 1

REMARQUE De tous les éléments structurants qui préservent l'homotopie, celui-ci est celui dont l'action est la plus proche d'un squelette (du fond, puisqu'il s'agit d'un épaisseur). Les clés *edge* := 1 et *grid* := 1 donnent des résultats visuellement plus satisfaisants.

PROCEDURE Mthick

SYNTAXE Mthick binin binout

OBJET Particularisation de l'épaissement **thick** à l'élément structurant M, i.e. :

		*	0			0	0	*
hexagonal	1	*	*	;	carré	*	*	1
		1	1			1	1	1

MODES grid : 1 , 0 edge : 1 , 0 depth : 1

REMARQUE Même type d'action que **Lthick**, mais avec beaucoup plus de barbules.

VOIR AUSSI **Lthick** et **Dthick**

PROCEDURE **Dthick**

SYNTAXE **Dthick** binin binout

OBJET Particularisation de l'épaississement **thick** à l'élément structurant D, i.e.:

			*	1			1	1	*
hexagonal	0	*	*	;	carré	*	*	0	
		0	0			0	0	0	

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1*

REMARQUE Pseudo-enveloppe convexe des grains. Réduit à un point les pores simplement connexes. Les résultats sont visuellement meilleurs lorsque la clé *edge* vaut 0. En hexagonal, permet d'exécuter, ou de démarrer, des épaississements à partir de composantes connexes réduites à un point.

VOIR AUSSI **Mthick**, **Lthick**

PROCEDURE **Lthin**

SYNTAXE **Lthin** binin binout

OBJET Dual de **Lthick** pour le complément.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1*

REMARQUE Amincissement de base pour simuler les squelettes. Donne des résultats particulièrement élégants en hexagonal, où forme et fond jouent des rôles symétriques

VOIR AUSSI **Mthin**, **Dthin**

PROCEDURE **Mthin**

SYNTAXE **Mthin** binin binout

OBJET Dual de **Mthick** pour le complément, utilise donc l'élément structurant de **Dthick**.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1*

REMARQUE Peu utilisé.

VOIR AUSSI **Lthin, Dthin**

PROCEDURE **Dthin**

SYNTAXE **Dthin** binin binout

OBJET Dual de **Dthick** pour le complément, utilise donc l'élément structurant de **Mthick**.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1*

REMARQUE Marqueur ponctuel des composantes simplement connexes. Résultats visuellement meilleurs quand *edge = 0*.

VOIR AUSSI **Lthin, Mthin**

PROCEDURE **endpoints**

SYNTAXE **endpoints** binin binout

OBJET Points extrémaux de l'image binaire binin.

MODES *grid : 1, 0* *edge : 1, [0]* *depth : 1*

REMARQUE Présente un intérêt particulier lorsque *binin* est le résultat d'un amincissement ou d'un épaissement homotopique .

PROCEDURE **mulpoints**

SYNTAXE **mulpoints** binin binout

OBJET Points multiples de l'image binaire *binin*.

MODES *grid : 1, 0* *edge : 1, [0]* *depth : 1*

REMARQUE Présente un intérêt particulier lorsque *binin* est le résultat d'un amincissement ou d'un épaissement homotopique .

PROCEDURE **delsp**

SYNTAXE **delsp** binin binout

OBJET Supprime les points isolés (points uniques) de l'image binaire *binin*. Place le résultat dans *binout*.

MODES *grid : 1, 0* *edge : 1, [0]* *depth : 1*

PROCEDURE gdscentre

SYNTAXE gdscentre binin binout

OBJET En *binout*, centre géodésique de l'image binaire *binin*

MODES *grid* : 1, [0] *edge* : 1, 0 *depth* : 1

REMARQUE Raffinement de la procédure **Dthin**, mais beaucoup plus longue et valable seulement en trame hexagonale .

11.2 Amincissements, épaissements numériques

PROCEDURE **greyseero**

SYNTAXE **greyseero** el_struct greyin greyout

OBJET Erosion de *greyin* par un sous-ensemble quelconque el_struct du carré ou de l'hexagone unité, placée dans *greyout*.

MODES *grid : 1, 0* *edge : 1* *depth : 8, 16*

PROCEDURE **greysedil**

SYNTAXE **greysedil** el_struct greyin greyout

OBJET Dilatation duale de **greyseero** vis à vis du passage au négatif.

MODES *grid : 1, 0* *edge : 1* *depth : 8, 16*

PROCEDURE **greythinstep**

SYNTAXE **greythinstep** se_for_inf se_for_sup greyin greyout

OBJET Passe d'amincissement (numérique) de *greyin* par l'élément structurant mixte { se_for_inf ; se_for_sup }, placée dans *greyout*.

MODES *grid : 1, 0* *edge : 1* *depth : 8, 16*

PROCEDURE **greythickstep**

SYNTAXE **greythickstep** se_for_inf se_for_sup greyin greyout

OBJET Epaissement dual de **greythinstep** pour le passage au négatif.

MODES *grid : 1, 0* *edge : 1* *depth : 8, 16*

PROCEDURE **dirgradient**

SYNTAXE **dirgradient** dir greyin greyout

OBJET Gradient dans la direction *dir* en trame hexagonale, défini par la différence entre **greythickstep** et **greythinstep**, appliqués à greyin avec pour éléments structurants les deux points +1 et -1 dans la direction *dir*.

MODES *grid : 1* *edge : 1* *depth : 8, 16*

REMARQUE L'utilisation d'épaississement et d'amincissement au lieu de dilatation et d'érosion permet à la procédure dérivée **gradvect** d'obtenir 12 directions d'azimut.

PROCEDURE **vectgrad0**

SYNTAXE **vectgrad0** greyin greyout1 greyout2

OBJET Gradient complet brut de *greyin* obtenu par application de **dirgradient** dans les six directions hexagonales. Le module est placé dans *greyout1* et l'azimut dans *greyout2*.

MODES *grid : 1* *edge : 1* *depth : 8, 16*

REMARQUE Des directions multiples ou incohérentes peuvent apparaître (cf. **gradvect**)

PROCEDURE **gradvect**

SYNTAXE **gradvect** greyin greyout1 greyout2

OBJET Gradient vectoriel vrai obtenu par tri des directions sur **gradvect0**. Le module est placé dans *greyout1* et l'azimut sur 12 valeurs, plus le zéro, dans *greyout2*. Les valeurs paires correspondent aux directions de la trame, les impaires à la trame conjuguée.

MODES *grid : 1* *edge : 1* *depth : 8, 16*

PROCEDURE **mainthin**

SYNTAXE **mainthin** binin binout sz

OBJET Amincit *binin* en utilisant la procédure **Lthin**, puis supprime les barbules dont la longueur est inférieure à *sz*.

MODES *grid : 1,0* *edge : 1,0* *depth : 1*

VOIR AUSSI **Lthin**, **clip**

PROCEDURE **greyhmt**

SYNTAXE **greyhmt** imin es1 es2 imout

OBJET Transformée en tout-ou-rien pour l'image 8-bit *imin*, par les deux éléments structurants: *es1* (pour les dômes) et *es2* (pour les vallées).

MODES *grid* : 1,0 *edge* : 1,0 *depth* : 8

VOIR AUSSI **greythickstep, greythinstep**

PROCEDURE **greythickturn**

SYNTAXE **greythickturn** es1 es2 imin imout

OBJET Un cycle de six (*grid*=1) ou huit (*grid*=0) pas élémentaires d'amincissement numérique (i.e. la procédure *greythickstep*) selon les différentes directions de la grille.

MODES *grid* : 1,0 *edge* : 1,0 *depth* : 8

VOIR AUSSI **greythickstep, greythinstep**

PROCEDURE **greythick**

SYNTAXE **greythick** se1 se2 imin imout

OBJET Epaisseur à niveaux de gris par l'élément structurant *se1* (pour les dômes) et *se2* (pour les vallées), placé en *imout*.

MODES *grid* : 1,0 *edge* : 1,0 *depth* : 8

VOIR AUSSI **greythickstep, greythickturn, greythinstep**

12. LIGNE DE PARTAGE DES EAUX

PROCEDURE clip

SYNTAXE clip binin binout

OBJET Ebarbulage total de l'image *binin*, i.e. amincissement par

* 0 * 0 0

hexagonal * 1 0 ; carré 1 1 0

0 0 * 0 0

MODES grid : 1, 0 edge : 1, 0 depth : 1

REMARQUE Les bords sont traités différemment selon le choix de la clé *edge*.

PROCEDURE skiz

SYNTAXE skiz binin binout

OBJET En *binout*, amincissement homotopique totalement ébarbulé du complément de l'image binaire *binin*. Cette procédure, très utilisée en morphologie mathématique, se nomme "squelette par zones d'influence".

MODES grid : 1, 0 edge : 1, 0 depth : 1

REMARQUE Le résultat dépend de la clé *edge*. Par exemple, si *binin* est simplement connexe et strictement inclus dans le champ, pour *edge* = 1, on trouve en *binout* une courbe fermée à mi-distance entre *binin* et les bords du champ, alors que pour *edge* = 0 on trouve l'ensemble vide.

VARIANTE La variante géodésique (donnée seulement pour *grid* = 1 et *edge* = 1), est la procédure de syntaxe :

gdskiz binin binmask binout

PROCEDURE threshwshed

SYNTAXE threshwshed greyin binout

OBJET Ligne de partage des eaux non marquée de *greyin*, placée dans *binout*.

MODES grid : 1, 0 edge : 1 depth : 8, 16, to 1

REMARQUE La progression, ici arithmétique et de pas un, est très lente et fait que cette procédure n'est pratiquement utilisée qu'à des fins pédagogiques.

PROCEDURE **mwshed**

SYNTAXE **mwshed** *greyin* *binmark* *binout* *type*

OBJET Ligne de partage des eaux de *greyin* conditionnellement aux minima *binmark*, placée en *binout*. L'algorithme procède par seuils successifs en progression arithmétique (*type* = 1) ou géométrique (*type* = 0).

MODES *grid* : 1, 0 *edge* : 1 *depth* : 8, 16, to 1

REMARQUE Le *type* 0 convient mieux aux images telles que des gradients où les premiers seuils comptent particulièrement.

PROCEDURE **wshed**

SYNTAXE **wshed** *greyin* *binout1* *binout2* *type*

OBJET Ligne de partage des eaux de *greyin*, placée en *binout1*, les minima de *greyin* étant placés, eux, dans *binout2*. L'algorithme procède par seuils successifs en progression arithmétique (*type* = 1) ou géométrique (*type* = 0).

MODES *grid* : 1, 0 *edge* : 1 *depth* : 8, 16, to 1

13. SEGMENTATION

PROCEDURE **mgradwshed**

SYNTAXE **mgradwshed** greyin binmark binout type

OBJET Procédure **mwshed** appliquée à **gradient** de f , les paramètres *binmark*, *binout*, et *type* conservant la même définition que dans **mwshed**.

MODES *grid* : 1, 0 *edge* : 1 *depth* : 8, 16, to 1n

PROCEDURE **gradwshed**

SYNTAXE **gradwshed** greyin binout1 binout2 type

OBJET Procédure **wshed** appliquée à **gradient** de f , les paramètres *binout1*, *binout2* et *type* conservant la même définition que dans **wshed**.

MODES *grid* : 1, 0 *edge* : 1 *depth* : 8, 16, to 1

PROCEDURE **shapeseq**

SYNTAXE **shapeseq** greyin binin greyout size type

OBJET La procédure met à zéro (dans *greyout*) les points de *greyin* qui appartiennent à la L.P.E. du négatif de la fonction distance de *binin*. Pour éviter la sur-segmentation, la fonction distance est filtrée par ouverture par reconstruction, de taille *size* et de type *type* (si **diropen** *type*=1, si **open** *type*<>1).

MODES *grid* : 1, 0 *edge* : 1 *depth* : 8, 16 and 1

REMARQUE S'applique dans des situations similaires à celles de **cellseg**, et d'autant mieux que les zones à segmenter sont à peu près convexes.

PROCEDURE **smoothcnc**

SYNTAXE **smoothcnc** greyin k binout

OBJET La procédure génère sur l'image *greyin* une extraction de contrastes de telle façon que dans chaque composante connexe de l'image de sortie binaire *binout*, les fluctuations de gris dans n'importe quel hexagone de taille 2 soient inférieures à celles d'une parabole $y=kt^2$. La binarisation de *greyin* est basée sur la continuité régionale. *Binout* est formée des points x où la variation de *greyin* à l'intérieur de l'hexagone de taille 2 centré en x est inférieure à celle des paraboles kt^2 et $-kt^2$ (t est le niveau de gris).

MODES *grid* : 1, [0] *edge* : 1,0 *depth* : 8

VOIR AUSSI **jumpcnc, jump**

PROCEDURE **jumpcnc**

SYNTAXE **jumpcnc** *greyin* *binout* *k* *type*

OBJET La procédure génère sur l'image *greyin* une extraction de contrastes de telle façon que dans chaque composante connexe de l'image de sortie binaire *binout*, les fluctuations de gris se situent dans l'intervalle $2*k$.

Les composantes connexes sont obtenues:

si *type*=0 , à partir du minimum

si *type*=2 , à partir du maximum

si *type*=1, d'une manière auto-duale, à partir à la fois de minima et maxima.

MODES *grid* : 1, [0] *edge* : 1,0 *depth* : 8

VOIR AUSSI **smoothcnc**

PROCEDURE **mosaic**

SYNTAXE **mosaic** *greyin* *greyout1* *greyout2*

OBJET Image mosaïque de *greyin* (placée dans *greyout1*) et gradient de cette image mosaïque (placé dans *greyout2*). L'image mosaïque présente des zones plates correspondant aux bassins versants du gradient de l'image *greyin*.

MODES *grid* : 1, 0 *edge* : 1,0 *depth* : 8

VOIR AUSSI **wfall, kheops**

PROCEDURE **wfall**

SYNTAXE **wfall** *greyin* *greyout* *binout*

OBJET Algorithme des cascades appliqué à *greyin* (i.e. étape dans la construction de la pyramide par la procédure **kheops**).

MODES *grid : 1, 0* *edge : 1,0* *depth : 8*

EXEMPLE **mosaic, kheops**

PROCEDURE **kheops**

SYNTAXE **kheops** greyin greyout filename

OBJET Pyramide d'images mosaïque obtenue à partir de *greyin*, par des LPEs et des cascades successives. Les différentes images mosaïque sont visualisées dans *greyout* et sauvées dans le répertoire de Micromorph sous la forme "filename i.bmp" où i représente le i-ème niveau de la pyramide.

MODES *grid : 1, 0* *edge : 1,0* *depth : 8*

EXEMPLE **mosaic, wfall**

PROCEDURE **jump**

SYNTAXE **jump** greyin greyout jmp type

OBJET Segmente l'image de gris *greyin* par montée à partir des minima de hauteur *jmp* et par descente à partir des maxima de la même hauteur. Le processus est itéré. *Type* varie de 1 à 3 et permet un léger lissage pendant l'opération (3 est plus régulier que 1).

MODES *grid : 1, 0* *edge : 1,0* *depth : 8*

VOIR AUSSI **jumpcnc**

14. MESURES

En règle générale pour les fonctions de MICROMORPH, les valeurs ne sont que stockées dans une variable, puis affichées par le biais de **print**. Si l'on veut de plus les conserver, il faut faire précéder la fonction de l'ouverture d'un fichier et la faire suivre de la fermeture de celui-ci. Ainsi, si l'on veut connaître l'évolution du périmètre des érodés de l'image b1, on écrira:

```
output "c:\b1.dat"  
while (area b1) do  
print perim b1  
ero b1 b1 1  
end  
output " "
```

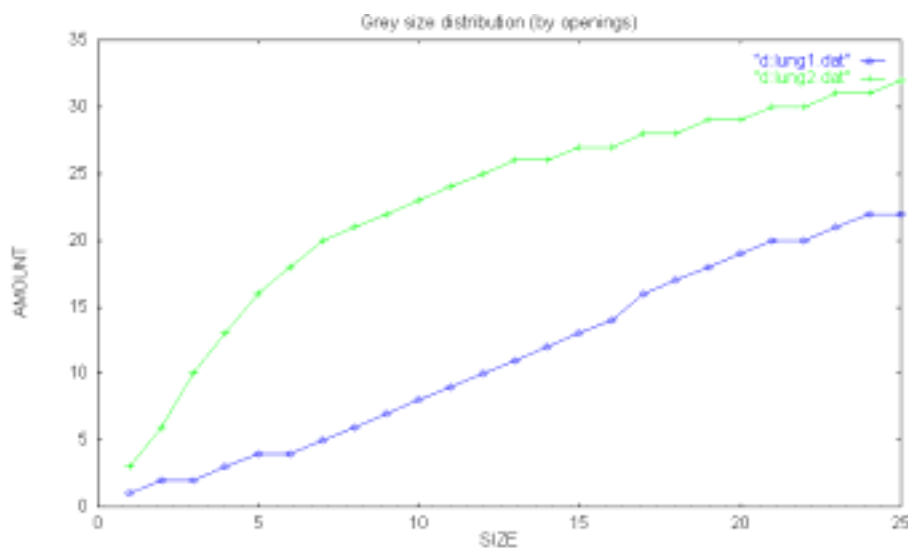
Toutefois, la plupart des fonctions de la section 15 (courbes) offrent un raccourci, car elles admettent un nom de fichier en variable d'entrée. Par exemple, pour la granulométrie des images lung1 et lung2, de 1 en 1 jusqu'au pas 25, on écrira, après avoir chargé les images en g1 :

```
granul g1 1 25 "lung1.dat"  
granul g1 1 25 "lung2.dat"
```

Les valeurs granulométriques sont alors enregistrées dans les fichiers *lung1.dat* et *lung2.dat*. Elles peuvent servir pour tout traitement ultérieur. En particulier, si l'on veut les représenter graphiquement, on pourra utiliser un programme externe comme par exemple *gnuplot*. Pour utiliser *gnuplot*, voyez le fichier d'aide de ce programme. En exécutant dans *gnuplot* la suite d'instructions suivantes :

```
set xlabel "SIZE" 0  
set ylabel "AMOUNT" 0  
set title "Grey size distribution (by openings)"  
plot "c:\lung1.dat" with linespoint , "c:\lung2.dat" with linespoint
```

on obtient le graphique suivant :



14.1 Mesures de base

Cette première section sur les mesures est consacré aux fonctionnelles stéréologiques binaires de base. Comme l'aire fait déjà partie des mots primitifs de Micromorph, sous le nom de **involume** (qui représente de plus l'intégrale des gris, en numérique), il reste la *variation diamétrale*, le *périmètre* (donné ici sous deux versions, l'une d'esprit digital l'autre euclidien), et le *nombre de connexité*, encore appelé constante d'EULER-POINCARÉ

FONCTION	diameter
SYNTAXE	diameter dir binin
OBJET	Renvoie le nombre des entrées dans les grains de s dans la direction <i>dir</i> .
MODES	<i>grid</i> : 1, 0 <i>edge</i> : 1, 0 <i>depth</i> : 1
REMARQUE	La fonction est indifférente au choix de la clé <i>edge</i> , et ne compte en aucun cas les intersections des grains avec les bords de champ comme des débuts d'intercepts. Fonctionne en trame carrée ou hexagonale
VOIR AUSSI	binhferret , binvferret

FONCTION	digperim
SYNTAXE	digperim binin binout
OBJET	Renvoie le nombre des éléments de contour <i>hexagonal</i> de l'ensemble <i>binin</i> . En prime, place en d l'ensemble des éléments de contour considérés
MODES	<i>grid</i> : 1, [0] <i>edge</i> : 1, 0 <i>depth</i> : 1
REMARQUE	Comme pour diameter , la clé <i>edge</i> est indifférente. digperim compte pour double les lignes d'épaisseur unité et ignore les points isolés (ce qui est un double différence avec l'aire de cont s). Par ailleurs, le contour digital de s étant distinct de celui de s ^c , la fonction digperim n'est pas auto-duale. On a cependant, lorsque s n'atteint pas les bords du champ :
	$\mathbf{digperim\ s} - \mathbf{digperim\ s^c} = 6 * \mathbf{cnumber\ s}$
VOIR AUSSI	cont , sq4cont , perim

FONCTION	perim
SYNTAXE	perim binin
OBJET	Renvoie le périmètre de <i>binin</i> estimé par la formule de CAUCHY.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1*

REMARQUE Mesure auto-duale, censée estimer sans biais le périmètre euclidien, lorsqu'il existe, à partir de données en trame carrée ou hexagonale.

VOIR AUSSI **cont** , **sq4cont**, **digperim**

FONCTION **cnumber**

SYNTAXE **cnumber** *binin* *tr*

OBJET Renvoie le nombre de connexité de l' image source, binaire, *binin*. La valeur binaire *tr* conditionne la clé *edge*. Si *tr* = 1, l'objet *s* est considéré comme un phénomène transitif, même s'il est limité par les bords de champ. Si *tr* = 0, on ne retient que les éléments structurants du dénombrement qui sont inclus dans le champ .

MODES *grid : 1, [0]* *edge : tr* *depth : 1*

REMARQUE La fonction bascule automatiquement en trame hexagonale le temps du calcul.

FONCTION **binhferret**

SYNTAXE **binhferret** *binin*

OBJET Diamètre de Ferret de l'ensemble *binin* dans la direction horizontale (i.e. variation diamétrale de l'enveloppe convexe de *binin*).

MODES *grid : 1* *edge : 1* *depth : 1*

REMARQUE Existe aussi en variante verticale (mêmes modes) sous le nom de **binvferret**

14.2 Mesures supplémentaires

FONCTION	var0
SYNTAXE	var0 <i>greyin</i>
OBJET	Variance d'ordre 0 de <i>greyin</i> , <i>i.e.</i> calcul de $E[f(x) - m]$, où $f(x)$ est la valeur de <i>greyin</i> au point courant x , et m sa moyenne.
MODES	<i>grid</i> : 1, 0 <i>edge</i> : 1, 0 <i>depth</i> : 8, 16

FONCTION	var1
SYNTAXE	var1 <i>greyin</i>
OBJET	Variance d'ordre 1 de $f = \text{greyin}$, <i>i.e.</i> calcul de $0.5E[f(x) - f(y)]$, où x et y parcourent <i>greyin</i> indépendamment l'un de l'autre. Algorithme : $\text{var1}(f) = \int_{\mathcal{T}} G(t) \cdot [1 - G(t)] dt$ où $G(t)$ est la fonction de répartition des teintes de f .
MODES	<i>grid</i> : 1, 0 <i>edge</i> : 1, 0 <i>depth</i> : 8, 16
REMARQUE	Si <i>greyin</i> est modélisé par une fonction aléatoire à variance d'ordre un finie, alors var1 (<i>greyin</i>) estime la valeur asymptotique de vario1 (<i>greyin</i>)

FONCTION	var2
SYNTAXE	var2 <i>greyin</i>
OBJET	Variance de <i>greyin</i> , <i>i.e.</i> calcul de $E[(f(x) - m)^2]$, où $f(x)$ est la valeur de <i>greyin</i> au point courant x , et m sa moyenne.
MODES	<i>grid</i> : 1, 0 <i>edge</i> : 1, 0 <i>depth</i> : 8, 16
REMARQUE	Si <i>greyin</i> est modélisé par une fonction aléatoire à variance finie, alors var2 (<i>greyin</i>) estime la valeur asymptotique de vario2 (<i>greyin</i>).

FONCTION	rug
SYNTAXE	rug <i>imin</i>
OBJET	Estime la rugosité de l'image binaire ou numérique <i>imin</i> , définie comme moyenne du carré de la courbure (de l'ensemble, ou des sections de la

fonction, selon le cas) en calculant la dérivée seconde à l'origine de la densité de distribution des intercepts. Interprétation stéréologique.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1, 8, 16

PROCEDURE **mse**

SYNTAXE *v* := **mse** *greyin1* *greyin2*

OBJET Retourne la différence quadratique moyenne entre les deux entrées.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 8

PROCEDURE **bincount**

SYNTAXE **bincount** *binin*

OBJET Retourne le nombre de composantes connexes de *binin*.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1

VOIR AUSSI **cnumber**

PROCEDURE **flatcount**

SYNTAXE **flatcount** *imin*

OBJET Retourne le nombre de zones plates dont l'érosion est non vide.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

PROCEDURE **flatzone**

SYNTAXE **flatzone** *imin*

OBJET Fonction retournant l'aire occupée par les zones plates de l'image d'entrée. Une zone est plate lorsqu'elle est connectée, quand son érosion élémentaire est non vide et quand son niveau de gris est constant.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 8, 16

15. COURBES

Toutes les procédures suivantes stockent les résultats des mesures dans un fichier dont le nom est donné dans les paramètres d'appel de chaque procédure. Ce fichier peut ensuite être utilisé pour tracer les courbes (par gnuplot par exemple) ou encore dans des tableurs.

FONCTION	bincov
SYNTAXE	bincov dir st spacing binin1 binin2 sizemax fname
OBJET	Covariance croisée entre les deux ensembles <i>binin1</i> et <i>binin2</i> dans la direction <i>dir</i> (en carré ou en hexagonal), de 0 à <i>sizemax</i> , par pas distants de <i>spacing</i> . La clé <i>st</i> permet de choisir entre représentation transitive (<i>st</i> = 0) ou intrinsèque (<i>st</i> = 1). Diviser par 1000 pour avoir la covariance.
MODES	<i>grid</i> : 1, 0 <i>edge</i> : 1, 0 <i>depth</i> : 1
REMARQUE	On obtient la covariance usuelle de <i>binin1</i> en prenant <i>binin2</i> = <i>binin1</i> , et son variogramme en prenant <i>binin2</i> = (<i>binin1</i>) ^c .
VOIR AUSSI	vario1, vario2, covar

PROCEDURE	covar
SYNTAXE	covar dir spacing imin sizemax fname
SYNTAXE	Covariance de l'image de gris <i>greyin</i> dans la direction <i>dir</i> (définie en trame carrée ou hexagonale selon la valeur de <i>grid</i>), depuis l'abscisse 0 jusqu'à <i>sizemax</i> , par pas de valeur <i>spacing</i> . Les résultats sont stockés dans le fichier <i>fname</i> . Le nom du fichier <i>fname</i> doit être écrit entre guillemets
MODES	<i>grid</i> : 1, 0 <i>edge</i> : 1, [0] <i>depth</i> : 8, 16
EXEMPLE	covar 2 1 3 30 "c:\toto.dat" calculé la covariance de l'image 2 par pas de 3, de la valeur 0 à la valeur 30, dans la direction nord (si <i>grid</i> = 0) ou nord-est (si <i>grid</i> = 1). Les résultats sont placés dans c:\toto.dat .
VOIR AUSSI	vario2, bincov

FONCTION	vario1
SYNTAXE	vario1 dir st spacing greyin sizemax fname
OBJET	Variogramme d'ordre 1 de l' image numérique <i>greyin</i> , <i>i.e.</i> demi-espérance du module de $f(x+h) - f(x)$, dans la direction <i>dir</i> (en carré ou en hexagonal) de

0 à *sizemax*, par pas distants de *spacing*. La clé *st* permet de choisir entre représentation transitive (*st* = 0) ou intrinsèque (*st* = 1). Diviser par 10 pour avoir le variogramme d'ordre 1.

MODES *grid* : 1, 0 *edge* : 1, [0] *depth* : 8;16

REMARQUE Le palier de **vario1**, lorsqu' il existe, s'ajuste à l'aide de la variance "**var1** greyin".

FONCTION **vario2**

SYNTAXE **vario2** dir st spacing greyin sizemax fname

OBJET Variogramme d'ordre 2, *i.e.* demi-variance des accroissements $f(x+h) - f(x)$ dans la direction *dir*, de l'image numérique *greyin* (en carré ou en hexagonal), de 0 à *sizemax*, par pas distants de *spacing*. La clé *st* permet de choisir entre représentation transitive (*st* = 0) ou intrinsèque (*st* = 1).

OBJET *grid* : 1, 0 *edge* : 1, [0] *depth* : 8, 16

REMARQUE Le palier de **vario2**, lorsqu'il existe, s'ajuste à l'aide du double de la variance "**var2** greyin". Dans ce cas, *et dans ce cas seulement*, on peut parler de covariance, et elle vaut alors la variance de *greyin* diminuée du variogramme d'ordre 2. Sinon, la covariance n'existe pas (c'est pour cela qu'elle n'a pas été programmée directement).

FONCTION **modcont**

SYNTAXE **modcont** greyin size

OBJET Module de continuité de *greyin*, en trame carrée, ou encore variogramme d'ordre ∞ . Posant C pour le carré 3×3 , il vient au pas k :

$$\mathbf{modcont} \text{ greyin} := (\text{greyin} - \text{greyin} \ominus kC) \vee (\text{greyin} \oplus kC - \text{greyin})$$

MODES *grid* : 0 *edge* : 1 *depth* : 8, 16

FONCTION **isogranul**

SYNTAXE **isogranul** imin sp sizemax fname

OBJET Granulométrie isotrope binaire ou numérique. Le programme passe automatiquement en trame carrée le temps du calcul, qui ne se déroule qu'en mode standard, selon les mêmes pondérations que pour **granul**.

MODES *grid* : [1], 0 *edge* : 1 *depth* : 1, 8, 16

REMARQUE Versions dédoublées : **bisogranul** et **gisogranul**

VOIR AUSSI **granul**

FONCTION **granul**

SYNTAXE **granul** imin sp sizemax fname

OBJET Granulométrie de imin, binaire ou numérique (selon la profondeur d'imin), en mode standard ou intrinsèque, en trame carrée ou hexagonale. La progression du pas, au delà de 1, vaut sp, et la taille maximale est sizemax. En mode standard, il vient :

granul (i) = 1000 [1 - **area** (binin o B_i) / **area** binin] en binaire,
granul (i) = 100 [1 - **volume** (greyin o B_i) / **volume** greyin] en num.

où B_i désigne le carré ou l'hexagone de taille i. Lorsque imin (ou en numérique, son support) ne rencontre pas les bords du champ, le mode standard équivaut à la version transitive de la granulométrie. En mode intrinsèque, i.e. *ici* quand *edge* = 0, on a :

granul (i) = 1000 [1 - **area** (binin o B_i) / **area** (champ ⊖ B_{2i})], *bin*
granul (i) = 100 [1 - **volume** (greyin o B_i) / **area** (champ ⊖ B_{2i})], *num*

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

REMARQUE Les anti-granulométries, par fermeture, se construisent par dualité à partir de **granul**. Versions dédoublées : **bingranul** et **greygranul**.

PROCEDURE **pvonl**

SYNTAXE **pvonl** dir spc szmax fname

OBJET Retourne la proportion de surface des érodés de *binin* selon des segments pris dans la direction *dir*, et de longueur variant par pas de *spc* depuis 0 jusqu'à *szmax*. Les résultats sont stockés dans le fichier *fname*, qui est créé par la procédure et placé dans le répertoire *wmmorph*. Le nom du fichier *fname* doit être écrit entre guillemets

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1

EXEMPLE **pvonl** 2 1 3 30 "toto.dat"

 calculé l'érodé linéaire de l'image 2 par pas de 3, de la valeur 0 à la valeur 30, dans la direction nord (si *grid* = 0) ou nord-est (si *grid* = 1). Les résultats sont placés dans C :\wmmorph\toto.dat .

VOIR AUSSI **p1vonl**, **p2vonl**

PROCEDURE **p1vonl**

SYNTAXE **p1vonl** dir spc szmax fname

OBJET Retourne l'histogramme cumulé des intercepts de *binin*, pondéré en nombre, selon des segments pris dans la direction *dir*, et de longueur variant par pas de *spc* depuis 0 jusqu'à *szmax*. Les résultats sont stockés dans le fichier *fname*, qui est créé par la procédure et placé dans le répertoire *wmmorph*. Le nom du fichier *fname* doit être écrit entre guillemets.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1

EXEMPLE **p1vonl** 2 1 3 30 "toto.dat "

calcule l'histogramme cumulé des intercepts de l'image 2 par pas de 3, de la valeur 0 a la valeur 30, dans la direction nord (si *grid* =0) ou nord-est (si *grid* = 1). Les résultats sont placés dans C:\wmmorph\toto.dat .

VOIR AUSSI **pvonl, p2vonl**

PROCEDURE **p2vonl**

SYNTAXE **p2vonl** dir spc szmax fname

OBJET Retourne l'histogramme non cumulé des intercepts de *binin*, pondéré en nombre, selon des segments pris dans la direction *dir*, et de longueur variant par pas de *spc* depuis 0 jusqu'à *szmax*. Les résultats sont stockés dans le fichier *fname*, qui est créé par la procédure et placé dans le répertoire *wmmorph*. Le nom du fichier *fname* doit être écrit entre guillemets.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1

EXEMPLE **p2vonl** 2 1 3 30 "toto.dat"

calcule l'histogramme non cumulé des intercepts de l'image 2 par pas de 3, de la valeur 0 a la valeur 30, dans la direction nord (si *grid* =0) ou nord-est (si *grid* = 1). Les résultats sont placés dans C :\wmmorph\toto.dat

VOIR AUSSI **pvonl, p1vonl**

16. SIMULATIONS ALEATOIRES

16.1 Simulations Booléennes

PROCEDURE **point**

SYNTAXE **point** *imout*

OBJET Place un point au milieu de *imout* (de niveau de gris égal à 100 dans le cas numérique).

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1, 8, 16*

PROCEDURE **points**

SYNTAXE **points** *imout* *n*

OBJET Simule *n* points poissonniens dans *imout* (de niveau de gris égal à 100 dans le cas numérique).

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1, 8, 16*

PROCEDURE **regpoints**

SYNTAXE **regpoints** *greyin* *binout* *grey_spacing*

OBJET Simule des points Poissonniens de densité régionalisée par les intensités de *greyin*. Le paramètre positif *grey_spacing* indique l'intervalle entre les niveaux de gris pris en compte, et commande aussi le nombre des points. Il convient de le faire varier entre 1 et 10.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1*

PROCEDURE **isobool**

SYNTAXE **isobool** *binout* *size* *n*

OBJET Simule un ensemble Booléen avec *n* germes et pour *grain* primaire le disque de rayon *size*.

MODES *grid : 1, 0* *edge : [1], 0* *depth : 1*

PROCEDURE **isobool2**

SYNTAXE **isobool2** binout size n

OBJET Simule un ensemble Booléen avec $100 \times n$ germes, et avec pour grain primaire le disque de rayon aléatoire $k \times size$, où k est tiré dans la loi de Poisson de paramètre 2.

MODES *grid* : 1, 0 *edge* : [1], 0 *depth* : 1

EXEMPLE Valeurs typiques : *size* = 3 ; *n* = 2 .

PROCEDURE **eldil**

SYNTAXE **eldil** dir binin binout size

OBJET Dilate l'ensemble *binin* par une ellipse de taille *size* et de direction *dir*, et le place dans *binout*. Les tailles standard vont de 1 à 3.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1

VARIANTE **dropdil**, où l'élément structurant est une gouttelette dissymétrique.

PROCEDURE **elbool**

SYNTAXE **elbool** binout size n1 n2

OBJET Simule la réunion de deux ensembles booléens ayant pour grains primaires des ellipses de **eldil**, de taille *size*, horizontales et en nombre *n1* pour le premier, verticales et en nombre *n2* pour le second.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1

REMARQUE Faire varier *size* de 1 à 2, pour ne pas trop envahir l'espace.

VARIANTE **dropbool**, où les grains primaires sont les gouttelettes de **dropdil** (de taille 1 ou 2) **tribool** (en hexagonal) à grains primaires triangulaires (tailles utiles de 1 à 30).

PROCEDURE **hard**

SYNTAXE **hard** binout1 binout2 size1 size2 n1 n2

OBJET Simulation hiérarchique semblable à **rose**, mais où le second ensemble est astreint à rester disjoint du premier, au lieu de s'agglutiner à lui.

MODES *grid* : [1], 0 *edge* : 1 *depth* : 1

EXEMPLE Typiquement : $size1 = 10$; $size2 = 1$; $n1 = 500$; $n2 = 70$.

PROCEDURE **rose**

SYNTAXE **rose** binout1 binout2 size1 size2 n1 n2

OBJET Simulation de la hiérarchie de deux ensembles booléens, où le second est réduit aux grains qui rencontrent le premier. Celui-ci est du type **isobool** $size1\ n1$ ", le second est la réunion de deux "**elbol** $size2\ n2\ n2$ ", de directions horizontale et verticale. Cette réunion est placée dans *binout2*, et jointe à **isobool** dans *binout1*.

MODES *grid* : [1], 0 *edge* : 1 *depth* : bin

EXEMPLE Typiquement : $size1 = 15$; $size2 = (1\ \text{ou}\ 2)$; $n1 = 50$; $n2 = 150$

PROCEDURE **disjoint**

SYNTAXE **disjoint** binin binout size n

OBJET Simulation de disques de rayon *size*, disjoints entre eux et disjoints de *binin*. *Binout* contient la réunion de ces disques et de *binin*. Il est procédé à *n* tirages successifs de centres d'implantations uniformes; chacun n'est maintenu que si le disque correspondant ne rencontre pas ceux qui le précèdent.

MODES *grid* : 1, 0 *edge* : 1 *depth* : bin

EXEMPLE Typiquement : $size = 15$; $n = 150$.

PROCEDURE **flake**

SYNTAXE **flake** binout size n st

OBJET Version aléatoire (et non autohomothétique ...) du célèbre flocon de neige de Von Koch, par une technique qui généralise la procédure **rose**. On plante d'abord *n* points de Poisson qu'on dilate par le disque de rayon *size*; puis on tire $n + st$ points de Poisson. Ceux qui ne rencontrent pas le dilaté précédent sont éliminés, les autres subissent une dilatation circulaire de taille $size - 1$. Et l'on itère jusqu'à la taille 1.

MODES *grid* : 1, 0 *edge* : 1 *depth* : bin

EXEMPLE Domaines de variation : $size = \{ 5\ \text{à}\ 25 \}$; $n = \{ 2\ \text{à}\ 6 \}$; $st = \{ 10\ \text{à}\ 60 \}$. Typiquement : 18, 6, 30.

PROCEDURE conebool

SYNTAXE conebool greyout size n

OBJET Simulation de cônes booléens hexagonaux, de pente unité et de hauteur *size*, au nombre de *n*.

MODES *grid* : 1, [0] *edge* : 1 *depth* : 8, 16

REMARQUE Il faut garder *n* inférieur ou égal à 250.

VARIANTE conebool2, où les grains primaires sont des cônes à base dodécagonale.

PROCEDURE rocky

SYNTAXE rocky greyout greydisplay (shadow of greyout) width n type

OBJET Simulation de la fonction aléatoire booléenne des fonds rocheux. Les cônes, de pente unité, ont pour bases soit des hexagones (*type*=1) soit des dodécagones (*type* = 0). Les cotes de leurs sommets sont uniformément réparties entre 0 et *width*, avec pour densité *n* à chaque niveau.

MODES *grid* : 1, [0] *edge* : 1 *depth* : 8,16

EXEMPLE Valeurs typiques : *width* = 10, *n* = 4

16.2 Droites et Partitions

PROCEDURE **lines**

SYNTAXE **lines** binout n

OBJET Simulation de droites poissonniennes horizontales et verticales au nombre de n dans chacune des deux directions.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1

PROCEDURE **diags**

SYNTAXE **diags** binout n

OBJET Simulation de droites poissonniennes de densité isotrope n dans chacune des deux directions diagonales du carré et des quatre directions du cheval sur l'échiquier.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1

PROCEDURE **steps**

SYNTAXE **steps** dir greyout n

OBJET Simulation de bandes poissonniennes horizontales ($dir = 3$), ou verticales ($dir = 1$), variant par sauts de valeur 10 à chaque frontière.

MODES *grid* : [1], 0 *edge* : 1 *depth* : 8, 16

REMARQUE Pour éviter les overflows, prendre $n \leq 24$.

PROCEDURE **nestlines**

SYNTAXE **nestlines** binout n1 n2 k

OBJET Simule des cloisonnements poissonniens rectangulaires. Effectue une première simulation de **lines** d'intensité $n1$, dont certains des rectangles sont choisis avec la probabilité $1/k$ (k entier). Dans chaque polygone retenu, simule une réalisation différente de **lines**, d'intensité $n2$, et prend leur réunion.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1

EXEMPLE Valeurs typiques : $n2 = 10$. $n1$ $k = 10$

17. GRAPHERS

PROCEDURE **Gdil**

SYNTAXE **Gdil** greyin labels greyout size

OBJET Génère la dilatation de taille *size* de l'image *greyin*, considérée comme un graphe planaire d'étiquetage *labels*.

REMARQUE Le graphe transformé a pour valeurs *greyout*, mais conserve comme étiquetage *labels*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8*

VOIR AUSSI **Gero, Gopen, Gclose, Gbuild, label**

PROCEDURE **Gero**

SYNTAXE **Gero** greyin labels greyout size

OBJET Génère l'érosion de taille *size* de l'image *greyin*, considérée comme un graphe planaire d'étiquetage *labels*.

REMARQUE Le graphe transformé a pour valeurs *greyout*, mais conserve comme étiquetage *labels*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8*

VOIR AUSSI **Gdil, Gopen, Gclose, Gbuild, label**

PROCEDURE **Gopen**

SYNTAXE **Gopen** greyin labels greyout size

OBJET Génère l'ouverture de taille *size* de l'image *greyin*, considérée comme un graphe planaire d'étiquetage *labels*.

REMARQUE Le graphe transformé a pour valeurs *greyout*, mais conserve comme étiquetage *labels*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8*

VOIR AUSSI **Gero, Gdil, Gclose, Gbuild, label**

PROCEDURE **Gclose**

SYNTAXE **Gclose** greyin labels greyout size

OBJET Génère la fermeture de taille *size* de l'image *greyin*, considérée comme un graphe planaire d'étiquetage *labels*.

REMARQUE Le graphe transformé a pour valeurs *greyout*, mais conserve comme étiquetage *labels* .

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8*

VOIR AUSSI **Gero, Gdil, Gopen, Gbuild, label**

PROCEDURE **setborder**

SYNTAXE **setborder** imin imout

OBJET Trace une frontière fine, de valeur numérique 0, entre les objets de différents niveaux de gris.

REMARQUE Le graphe transformé a pour valeurs *greyout*, mais conserve comme étiquetage *labels*

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8*

PROCEDURE **Gbuild**

SYNTAXE **Gbuild** greyin labelin binmask greyout labelout

OBJET Restriction du graphe planaire (*greyin, labelin*) à ses composantes qui rencontrent *binmask*. Les autres composantes ne sont pas différenciées, elles sont considérées comme l'extérieur du graphe et regroupées sous l'étiquette de valeur zéro .

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8*

VOIR AUSSI **Gero, Gdil, Gopen, Gclose, label**

PROCEDURE **label**

SYNTAXE **label** greyin greyout

OBJET Engendre une image d'étiquettes associées à *greyin*, ce qui permet de traiter cette dernière comme un graphe planaire. Les maxima de *greyin* sont

affectés de la valeur 1, puis les maxima de *greyin* diminuée de ses maxima sont affectés de la valeur 2, et ainsi de suite.

SYNTAXE

1. Chaque étiquette est positive. La valeur zéro peut éventuellement être utilisée pour l'extérieur du graphe.

2. Dans toutes les procédures **Gdil**, **Gero**, **Gopen**, **Gclose** et **Gbuild**, portant sur un graphe, il faut toujours introduire à la fois une image numérique et un étiquetage associé.

MODES

grid : 1, 0

edge : 1, 0

depth : 8

VOIR AUSSI

Gero, **Gdil**, **Gopen**, **Gclose**, **Gbuild**

18. UTILITAIRES 3D

PROCEDURE **Seqload**

SYNTAXE **Seqload** SeqIn SeqIdStart SeqMemStart NbOfImages

OBJET Charge une séquence de *NbOfImages* dans la mémoire de travail de micromorph. Les images à charger doivent être dénommées [*SeqIn* *i*] (cf. exemple ci-dessous), où *i* varie de *SeqIdStart* à (*SeqIdStart* + *NbOfImages* - 1). Les images sont transférées dans les plans mémoire allant de *SeqMemStart* à (*SeqMemStart* + *NbOfImages* - 1). Tous ces plans mémoire doivent avoir été alloués au préalable (par la procédure **imalloc**), et ils doivent avoir la même profondeur de bits que les images à charger.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8

REMARQUE Lorsqu'on lance micromorph, le fichier d'initialisation *init.mic* alloue automatiquement 10 mémoires de 8 bits, puis 10 mémoires de 1 bit, puis 3 mémoires de 16 bits. Cette allocation automatique se fait de plus à chaque utilisation de la procédure format. Par conséquent, quand on alloue des mémoires supplémentaires, celles-ci commencent à partir du numéro d'ordre 24. Par exemple, le programme:

```
int i;
i := 1
for 1 to 12 do
i := imalloc 8
++ i
end
```

crée 12 allocations de mémoires de 8 bits, allant des numéros d'ordre 24 à 35. Si maintenant on exécute la procédure:

```
Seqload "c:\wmmorph\seq\toto_" 10 24 7
```

Cela revient à avoir exécuté les sept procédures:

```
imload 24 "c:\wmmorph\seq\toto_10.bmp"
```

```
.....
```

```
imload 30 "c:\wmmorph\seq\toto_16.bmp"
```

(et il reste encore 5 mémoires de disponible).

REMARQUE

EXEMPLE **Seqload** "c:\wmmorph\seq\toto_" 10 25 50

charge 50 images dans la mémoire de l'ordinateur:

```
imload 25 "c:\wmmorph\seq\toto_10.bmp"
```

```
imload 26 "c:\wmmorph\seq\toto_11.bmp"
```

```
...
```

```
...
```

```
imload 74 "c:\wmmorph\seq\toto_59.bmp"
```

VOIR AUSSI **Seqsave**

PROCEDURE **Seqsave**

SYNTAXE **Seqsave** SeqIn SeqMemStart SeqStart NbOfImages

OBJET Sauvegarde sur le disque la séquence de *NbOfImages* commençant à la mémoire *SeqStart*. Les images stockées sont dénommées [*SeqIn* *i*] (cf. exemple ci-dessous), où *i* varie de *SeqldStart* à (*SeqldStart* + *NbOfImages* -)

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8

EXEMPLE La procédure:

```
Seqsave "c:\wmmorph\seq\toto_" 24 10 7
```

Equivaut aux sept instructions:

```
imsave 24 "c:\wmmorph\seq\toto_ 10.bmp"  
.....  
imsave 30 "c:\wmmorph\seq\toto_ 16.bmp"
```

VOIR AUSSI **seqload**

PROCEDURE **Seqclr**

SYNTAXE **Seqclr** SeqInStart NbOfImages

OBJET Efface les images en nombre *NbOfImages* et commençant à la mémoire *SeqInStart*.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE La procédure : **Seqclr** 25 50

efface les 50 images placées dans les mémoires 25 à 74.

VOIR AUSSI **Seqload**

PROCEDURE **Seqset**

SYNTAXE **Seqset** Seqoutstart NbOfImages value

OBJET Attribue une valeur constante à toutes les images depuis *SeqOutStart* jusqu'à *SeqOutStart* + *NbOfImages* - 1.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1, 8, 16*

VOIR AUSSI **Seqload**

PROCEDURE **Seqcopy**

SYNTAXE **Seqcopy** SeqInStart SeqOutStart NbOfImages

OBJET Copie les *NbOfImages* plans de mémoire commençant en *SeqInStart* dans le plan de mémoire commençant en *SeqOutStart*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1, 8, 16*

EXEMPLE **Seqcopy** 25 75 50

copie l'image 25 dans le plan 75, la 26 dans le plan 76, etc.. jusqu'à l'image 74, placée en plan 124 (les plans 75-124 doivent avoir été alloués au préalable, cf. **Seqload**).

VOIR AUSSI **Seqload**

PROCEDURE **Seqinv**

SYNTAXE **Seqinv** SeqInStart SeqOutStart NbOfImages

OBJET Inverse chaque image de la séquence qui débute en *SeqInStart* et la place dans la séquence qui débute en *SeqOutStart*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1, 8, 16*

EXEMPLE **Seqinv** 25 75 50

inverse l'image 25 et la place dans le plan 75, inverse la 26 et la place dans le plan 76, etc.. jusqu'à l'image 74, placée en plan 124 (les plans 75-124 doivent avoir été alloués au préalable, cf. **Seqload**).

PROCEDURE **Seqadd**

SYNTAXE **Seqadd** SeqInStart ImOut NbOfImages

OBJET Somme les *NbOfImages* images commençant en *SeqInStart*, et place la somme dans l'image *Imout* (qu'il faudra souvent prendre de profondeur égale à 16 bits).

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1, 8, 16*

EXEMPLE **Seqadd** 25 100 50

Place en mémoire n° 100 la somme (image 25 + image 26 +...+ Image 74).

VOIR AUSSI **Seqcadd, Seqseqadd**

PROCEDURE **Seqcadd**

SYNTAXE **Seqcadd** SeqInStart Const SeqOutStart NbOfImages

OBJET Ajoute la constante *Const* à chaque image de la séquence qui débute en *SeqInStart* et place les résultats dans la séquence qui débute en *SeqOutStart*. Les plans de mémoire depuis *SeqOutStart* jusqu'à (*SeqOutStart* + *NbOfImages* - 1) doivent avoir été alloués au préalable (cf. **Seqload**).

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **Seqcadd** 25 9 75 50

Ajoute aux 50 images de la séquence la valeur 9 et la place les résultats dans les 50 plans commençant à partir de 75 (les plans 75-124 doivent avoir été alloués au préalable, cf. **Seqload**) .

VOIR AUSSI **Seqcsub, Seqadd, Seqseqadd**

PROCEDURE **Seqseqadd**

SYNTAXE **Seqseqadd** SeqIn1Start SeqIn2Start SeqOutStart NbOfImages

OBJET Ajoute terme à terme chacune des *NbOfImages* images de la séquence qui débute en *SeqIn1Start* à chaque image de la séquence qui débute en *SeqIn2Start* et place les résultats dans la séquence qui débute en *SeqOutStart*. Les plans de mémoire depuis *SeqOutStart* jusqu'à (*SeqOutStart* + *NbOfImages* - 1) doivent avoir été alloués au préalable (cf. **Seqload**).

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **Seqseqadd** 30 50 100 20

Ajoute terme à terme les 20 images de la séquence commençant en 30 à celles de la séquence commençant en 50, et la place les résultats dans les 20 plans commençant à partir de 100 (les plans 100-119 doivent avoir été alloués au préalable, cf. **Seqload**) .

VOIR AUSSI **Seqadd, Seqcadd, Seqseqsub**

PROCEDURE **Seqcsub**

SYNTAXE **Seqcsub** SeqInStart Const SeqOutStart NbOfImages

OBJET Retranche la constante *Const* à chaque image de la séquence qui débute en *SeqInStart* et place les résultats dans la séquence qui débute en *SeqOutStart*.

Les plans de mémoire depuis *SeqOutStart* jusqu'à (*SeqOutStart* + *NbOfImages* - 1) doivent avoir été alloués au préalable (cf. **Seqload**).

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **Seqsub** 25 9 75 50

retranche aux 50 images de la séquence la valeur 9 et la place les résultats dans les 50 plans commençant à partir de 75 (les plans 75-124 doivent avoir été alloués au préalable, cf. **Seqload**) .

VOIR AUSSI **Seqadd, Seqseqsub**

PROCEDURE **Seqseqsub**

SYNTAXE **Seqseqsub** SeqIn1Start SeqIn2Start SeqOutStart NbOfImages

OBJET Retranche terme à terme chacune des *NbOfImages* images de la séquence qui débute en *SeqIn1Start* à chaque image de la séquence qui débute en *SeqIn2Start* et place les résultats dans la séquence qui débute en *SeqOutStart*. Les plans de mémoire depuis *SeqOutStart* jusqu'à (*SeqOutStart* + *NbOfImages* - 1) doivent avoir été alloués au préalable (cf. **Seqload**).

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **Seqseqsub** 30 50 100 20 retranche terme à terme les 20 images de la séquence commençant en 30 à celles de la séquence commençant en 50, et la place les résultats dans les 20 plans commençant à partir de 100 (les plans 100-119 doivent avoir été alloués au préalable, cf. **Seqload**) .

VOIR AUSSI **Seqsub, Seqseqadd**

PROCEDURE **Seqcmul**

SYNTAXE **Seqcmul** Seqinstqrt Seqoutstqrt NbOfImages coeff

OBJET Multiplie par l'entier coeff toutes les images depuis *SeqInStart* et jusqu'à *SeqInStart* + *NbOfImages* - 1. Place les résultats en séquence dans les mémoires commençant en *SeqOutStart*.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

VOIR AUSSI **Seqcadd**

PROCEDURE **Seqmean**

SYNTAXE **Seqmean** SeqInStart ImOut NbOfImages

OBJET Effectue la moyenne arithmétique des *NbOfImages* images commençant en *SeqInStart*, et place le résultat dans l'image *Imout*.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **Seqadd** 25 100 50

Place en mémoire n° 100 la moyenne
(image 25 + image 26 +...+ image 74) / 50 .

VOIR AUSSI **Seqseqmean**

PROCEDURE **Seqseqmean**

SYNTAXE **Seqseqmean** SeqIn1Start SeqIn2Start SeqOutStart NbOfImages

OBJET Calcule la moyenne terme à terme de chaque paire des *NbOfImages* images des séquences qui débutent en *SeqIn1Start* et en *SeqIn2Start* respectivement, et place les résultats dans la séquence qui débute en *SeqOutStart*. Les plans de mémoire depuis *SeqOutStart* jusqu'à (*SeqOutStart* + *NbOfImages* - 1) doivent avoir été alloués au préalable (cf. **Seqload**).

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **Seqseqmean** 30 50 100 20

Fait la moyenne terme à terme entre les 20 images de la séquence commençant en 30 et celles de la séquence commençant en 50, et la place les résultats dans les 20 plans commençant à partir de 100 (les plans 100-119 doivent avoir été alloués au préalable, cf. **Seqload**).

VOIR AUSSI **Seqmean**

PROCEDURE **Seqdiff**

SYNTAXE **Seqdiff** SeqInStart ImRef SeqOutStart NbOfImages

OBJET Calcule le module de la différence entre l'image fixe *ImRef* et chaque image de la séquence qui débute en *SeqInStart*, et place les résultats dans la séquence qui débute en *SeqOutStart*. Les plans de mémoire depuis *SeqOutStart* jusqu'à (*SeqOutStart* + *NbOfImages* - 1) doivent avoir été alloués au préalable (cf. **Seqload**).

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **Seqdiff** 25 75 80 40

effectue les 40 opérations :

image 80 = | *image 25* - *image 75* |

image 81 = | *image 26* - *image 75* |

...

image 119 = | *image 64* - *image 75* |

PROCEDURE **Seqvolume**

SYNTAXE **Seqvolume** SeqInStart NbofImages Param

OBJET Calcule le volume de chaque image de la séquence *SeqInStart* et soustrait la valeur *Param* de chaque résultat. La fonction retourne la somme de l'opération pour les *NbofImages* images.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE La procédure **Seqvolume** 10 40 10000

effectue les opérations suivantes :

$a1 := \mathbf{involume}(\text{image } 10) - 10000$

.....

$A40 := \mathbf{involume}(\text{image } 49) - 10000$

$\text{Résultat} = (a1 + a2 + \dots + a40)$

REMARQUE Le paramètre param est introduit pour éviter un dépassement de capacité, car MicroMorph ne prend en compte que les entiers compris entre plus et moins 2.1E9.

PROCEDURE **Seqinf**

SYNTAXE **Seqinf** SeqInStart ImOut NbofImages

OBJET Calcule l'infimum des *NbofImages* images commençant en *SeqInStart*, et place le résultat dans l'image *Imout*.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **Seqinf** 10 80 40 effectue l'opération suivante:

$\text{image } 80 = \mathbf{iminf}(\text{image } 10, \text{image } 11, \dots, \text{image } 49)$

VOIR AUSSI **Seqseqinf**, **Seqsup**

PROCEDURE **Seqseqinf**

SYNTAXE **Seqseqinf** SeqIn1Start SeqIn2Start SeqOutStart NbofImages

OBJET Calcule l'inf terme à terme entre chacune des *NbofImages* images de la séquence qui débute en *SeqIn1Start* et chacune des images de la séquence qui débute en *SeqIn2Start*. Place les résultats dans la séquence qui débute en *SeqOutStart*. Les plans de mémoire depuis *SeqOutStart* jusqu'à (*SeqOutStart* + *NbofImages* - 1) doivent avoir été alloués au préalable (cf. **Seqload**).

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE La procédure **Seqseqinf** 10 50 100 40 effectue les opérations suivantes:

image100 = **iminf** (*image10* et *image50*)

image101 = **iminf** (*image11* et *image51*)

.....

image139 = **iminf** (*image49* et *image89*)

VOIR AUSSI **Seqinf, Seqseqsup**

PROCEDURE **Seqsup**

SYNTAXE **Seqsup** SeqInStart ImOut NbOfImages

OBJET Calcule le supremum des *NbOfImages* images commençant en *SeqInStart*, et place le résultat dans l'image *Imout* .

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

SYNTAXE **Seqsup** 10 80 40 effectue l'opération suivante:

image 80 = **imsup** (*image10*, *image11*, ..., *image49*)

VOIR AUSSI **Seqseqsup, Seqinf**

PROCEDURE **Seqseqsup**

SYNTAXE **Seqseqsup** SeqIn1Start SeqIn2Start SeqOutStart NbOfImages

OBJET Calcule le sup terme à terme entre chacune des *NbOfImages* images de la séquence qui débute en *SeqIn1Start* et chacune des images de la séquence qui débute en *SeqIn2Start*. Place les résultats dans la séquence qui débute en *SeqOutStart*. Les plans de mémoire depuis *SeqOutStart* jusqu'à (*SeqOutStart* + *NbOfImages* - 1) doivent avoir été alloués au préalable (cf. **Seqload**).

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE La procédure **Seqseqsup** 10 50 100 40 effectue les opérations suivantes:

image100 = **imsup** (*image10* et *image50*)

image101 = **imsup** (*image11* et *image51*)

.....

image139 = **imsup** (*image49* et *image89*)

VOIR AUSSI **Seqsup, Seqseqinf**

PROCEDURE Seqbuild

SYNTAXE **Seqbuild** SeqMaskStart SeqInOutStart NbOfImages

OBJET Effectue la reconstruction bi-dimensionnelle terme à terme des images de *SeqInOutStart* à partir des marqueurs commençant en *SeqMaskStart*, et place le résultat dans les mêmes emplacements mémoire que la séquence des images initiales. Les deux séquences doivent avoir *NbOfImages* images.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1, 8, 16*

EXEMPLE La procédure **Seqbuild** 10 80 40 effectue les opérations suivantes:

build 10 80
build 11 81
.....
build 49 119

PROCEDURE Seqthresh

SYNTAXE **Seqthresh** SeqInStart LoTh HiTh SeqOutStart NbOfImages

OBJET Seuille chaque image de la séquence d'images de gris commençant en *SeqInStart* et place les résultats séquentiellement dans les mémoires binaires commençant en *SeqOutStart*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 8, 16 to 1*

EXEMPLE La procédure *Seqthresh* 10 0 5 50 40 effectue les opérations suivantes:

imthresh 10 0 2 50
imthresh 11 0 2 51
....
imthresh 49 0 2 89

PROCEDURE Seqgrad

SYNTAXE **Seqgrad** SeqInStart SeqOutStart Size NbOfImages

OBJET Calcule le gradient de taille *Size* pour chacune des *NbOfImages* images commençant en *SeqInStart*, et place le résultat dans la séquence commençant en *SeqOutStart*.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1, 8, 16*

EXEMPLE La procédure **Seqgrad** 10 50 2 40 effectue les opérations suivantes:

gradient 10 50 2
gradient 11 51 2

PROCEDURE Seqhist

SYNTAXE **Seqhist** SeqInStart NbOfImages

OBJET Visualise la première image de la séquence (placée en *SeqInStart*), sur laquelle on peut cliquer en un point quelconque avec la souris. Affiche alors le profil des gris successifs le long de la séquence en ce point là. On peut répéter l'opération avec d'autres points. La procédure peut être arrêtée par les touches Ctrl +C.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1, 8, 16*

PROCEDURE Scroll

SYNTAXE **Scroll** SeqInStart NbOfImages temp NbTimes

OBJET Visualise le déroulement de la séquence commençant en *SeqInStart*, et de longueur *NbOfImages*. Le paramètre temp dose la vitesse de déroulement :

Temp = 0 => vitesse maximale (pas de temporisation)
Temp = 1 => visualisation pas à pas
Temp > 1 => l'intervalle de temps entre deux images successives augmente avec temp

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1, 8*

VOIR AUSSI **Scroll2, Scroll3, Visu**

PROCEDURE Scroll2

SYNTAXE **Scroll2** SeqIn1Start SeqIn2Start NbOfImages temp spacing NbTimes

OBJET Visualise le déroulement en parallèle des deux séquences commençant en *SeqIn1Start* et en *SeqIn2Start* respectivement, et de même longueur *NbOfImages*. Le paramètre temp dose la vitesse de déroulement :

Temp = 0 => vitesse maximale (pas de temporisation)
Temp = 1 => visualisation pas à pas
Temp > 1 => l'intervalle de temps entre deux images successives augmente avec temp

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1, 8*

VOIR AUSSI **Scroll, Scroll3, Visu**

PROCEDURE Scroll3

SYNTAXE **Scroll3** SeqIn1Start SeqIn2Start SeqIn3Start NbOfImages temp spacing NbTimes

SYNTAXE Visualise le déroulement en parallèle des trois séquences commençant en *SeqIn1Start*, en *SeqIn2Start* et en *SeqIn3Start* respectivement, et de même longueur *NbOfImages*. Le paramètre temp dose la vitesse de déroulement :

Temp = 0 => vitesse maximale (pas de temporisation)

Temp = 1 => visualisation pas à pas

Temp > 1 => l'intervalle de temps entre deux images successives augmente avec temp

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8

VOIR AUSSI **Scroll, Scroll2, Visu**

PROCEDURE Visu

SYNTAXE **Visu** SeqInStart NbOfImages greyout light k

OBJET Créé une vue en perspective de la séquence binaire qui commence en *SeqInStart* et se termine en *SeqInStart* + *NbOfImages* - 1, considérée comme la pile des coupes successives d'un matériau tridimensionnel. On peut la voir soit par le dessus (paramètre *k* = 1) soit par le dessous (paramètre *k* <> 1). Le niveau d'éclairément de la visualisation est réglable par le paramètre light, variable entre 0 et 100. L'image perspective apparaît en greyout.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1

VOIR AUSSI **Scroll, Scroll2, Scroll3**

19. PROCEDURES 3D

PROCEDURE **ero3D**

SYNTAXE **ero3D** SeqInStart SeqOutStart Size NbOfImages

OBJET Effectue l'érosion tridimensionnelle de taille *Size* pour la séquence de *NbofImages* commençant à la zone mémoire n° *SeqInStart*. Le résultat est placé dans la série des mémoires commençant au n° *SeqOutStart*.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **ero3D** 10 50 2 40

1) érode la séquence des images 10, 11, ..., 49 par un cube-octaèdre (*grid* = 1) ou par un cube (*grid* = 0) de taille 2 . Le résultat est placé dans les mémoires 50, 51, ..., 89.

VOIR AUSSI **dil3D**

PROCEDURE **dil3D**

SYNTAXE **dil3D** SeqInStart SeqOutStart Size NbOfImages

OBJET Effectue la dilatation tridimensionnelle de taille *Size* pour la séquence de *NbofImages* commençant à la zone mémoire n° *SeqInStart*. Le résultat est placé dans la série des mémoires commençant au n° *SeqOutStart*.

MODES *grid* : 1, 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **dil3D** 10 50 2 40

dilate la séquence des images 10, 11, ..., 49 par un cube-octaèdre (*grid* = 1) ou par un cube (*grid* = 0) de taille 2 . Le résultat est placé dans les mémoires 50, 51, ..., 89.

VOIR AUSSI **ero3D**

PROCEDURE **open3D**

SYNTAXE **open3D** SeqInStart SeqOutStart Size NbOfImages

OBJET Effectue l'ouverture tridimensionnelle de taille *Size* pour la séquence de *NbofImages* commençant à la zone mémoire n° *SeqInStart*. Le résultat est placé dans la série des mémoires commençant au n° *SeqOutStart*.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **open3D** 10 50 2 40

Effectue l'ouverture pour la séquence des images 10, 11, ..., 49 par un cube-octaèdre (*grid* = 1) ou par un cube (*grid* = 0) de taille 2 . Le résultat est placé dans les mémoires 50, 51, ..., 89.

VOIR AUSSI **close3D**

PROCEDURE **close3D**

SYNTAXE **close3D** SeqInStart SeqOutStart Size NbOfImages

OBJET Effectue la fermeture tridimensionnelle de taille *Size* pour la séquence de *NbOfImages* commençant à la zone mémoire n° *SeqInStart*. Le résultat est placé dans la série des mémoires commençant au n° *SeqOutStart*.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **close3D** 10 50 2 40

Effectue la fermeture pour la séquence des images 10, 11, ..., 49 par un cube-octaèdre (*grid* = 1) ou par un cube (*grid* = 0) de taille 2 . Le résultat est placé dans les mémoires 50, 51, ..., 89.

VOIR AUSSI **open3D**

PROCEDURE **gradient3D**

SYNTAXE **gradient3D** SeqInStart SeqOutStart Size NbOfImages

OBJET Effectue le gradient tridimensionnel de taille *Size* pour la séquence de *NbOfImages* commençant à la zone mémoire n° *SeqInStart*. Le résultat est placé dans la série des mémoires commençant au n° *SeqOutStart*.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **gradient3D** 10 50 2 40

Effectue le gradient pour la séquence des images 10, 11, ..., 49 par un cube-octaèdre (*grid* = 1) ou par un cube (*grid* = 0) de taille 2 . Le résultat est placé dans les mémoires 50, 51, ..., 89.

PROCEDURE build3D

SYNTAXE **build3D** SeqMaskStart SeqInOutStart NbofImages

OBJET Effectue la reconstruction tridimensionnelle pour la séquence de *NbofImages* commençant à la zone mémoire n° *SeqInStart*. Le résultat est placé dans la série des mémoires commençant au n° *SeqOutStart*.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **build3D** 10 50 40

Effectue la reconstruction pour la séquence des images 10, 11, ..., 49 par un cube-octaèdre (*grid* = 1) ou par un cube (*grid* = 0) de taille 2. Le résultat est placé dans les mémoires 50, 51, ..., 89.

PROCEDURE buildopen3D

SYNTAXE **buildopen3D** SeqInStart SeqOutStart Size NbofImages

OBJET Effectue l'ouverture par reconstruction tridimensionnelle de taille *Size* pour la séquence de *NbofImages* commençant à la zone mémoire n° *SeqInStart*. Le résultat est placé dans la série des mémoires commençant au n° *SeqOutStart*.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **buildopen3D** 10 50 2 40

Effectue l'ouverture par reconstruction pour la séquence des images 10, 11, ..., 49 par un cube-octaèdre (*grid* = 1) ou par un cube (*grid* = 0) de taille 2.

Le résultat est placé dans les mémoires 50, 51, ..., 89.

VOIR AUSSI **buildclose3D**

PROCEDURE buildclose3D

SYNTAXE **buildclose3D** SeqInStart SeqOutStart Size NbofImages

OBJET Effectue la fermeture par reconstruction tridimensionnelle de taille *Size* pour la séquence de *NbofImages* commençant à la zone mémoire n° *SeqInStart*. Le résultat est placé dans la série des mémoires commençant au n° *SeqOutStart*.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **buildclose3D** 10 50 2 40

Effectue la fermeture par reconstruction pour la séquence des images 10, 11, ..., 49 par un cube-octaèdre (*grid* = 1) ou par un cube (*grid* = 0) de taille 2 .
Le résultat est placé dans les mémoires 50, 51, ..., 89.

VOIR AUSSI **buildclose3D**

PROCEDURE **openth3D**

SYNTAXE **openth3D** SeqInStart SeqOutStart Size NbOfImages

OBJET Effectue le résidu de l'ouverture tridimensionnelle de taille *Size* pour la séquence de *NbofImages* commençant à la zone mémoire n° *SeqInStart*. Le résultat est placé dans la série des mémoires commençant au n° *SeqOutStart*.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **openth3D** 10 50 2 40

Effectue le résidu de l'ouverture pour la séquence des images 10, 11, ..., 49 par un cube-octaèdre (*grid* = 1) ou par un cube (*grid* = 0) de taille 2 .
Le résultat est placé dans les mémoires 50, 51, ..., 89.

VOIR AUSSI **closeth3D**

PROCEDURE **closeth3D**

SYNTAXE **closeth3D** SeqInStart SeqOutStart Size NbOfImages

OBJET Effectue le résidu de la fermeture tridimensionnelle de taille *Size* pour la séquence de *NbofImages* commençant à la zone mémoire n° *SeqInStart*. Le résultat est placé dans la série des mémoires commençant au n° *SeqOutStart*.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **closeth3D** 10 50 2 40

Effectue le résidu de la fermeture pour la séquence des images 10, 11, ..., 49 par un cube-octaèdre (*grid* = 1) ou par un cube (*grid* = 0) de taille 2 .
Le résultat est placé dans les mémoires 50, 51, ..., 89.

VOIR AUSSI **openth3D**

PROCEDURE **maxima3D**

SYNTAXE **maxima3D** SeqInStart SeqOutStart NbofImages

OBJET Calcule les maxima tridimensionnels de la séquence de *NbofImages* commençant à la zone mémoire n° *SeqInStart*. Le résultat (binaire) est placé dans la série des mémoires commençant au n° *SeqOutStart*.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 8, 16 to 1

VOIR AUSSI **minima3D**

PROCEDURE **minima3D**

SYNTAXE **minima3D** SeqInStart SeqOutStart NbofImages

OBJET Calcule les minima tridimensionnels de la séquence de *NbofImages* commençant à la zone mémoire n° *SeqInStart*. Le résultat (binaire) est placé dans la série des mémoires commençant au n° *SeqOutStart*.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 8, 16 to 1

VOIR AUSSI **maxima3D**

PROCEDURE **af3D**

SYNTAXE **af3D** SeqInStart SeqOutStart Size NbofImages Type

OBJET Calcule le filtre alterné de taille *Size*, et de base cube-octaédrique ou cubique pour la séquence de *NbofImages* commençant à la zone mémoire n° *SeqInStart*. Le résultat est placé dans la série des mémoires commençant au n° *SeqOutStart*. Si le paramètre *type* vaut 1, le filtre commence par la fermeture **close3D** suivie de **open3D**, sinon c'est l'inverse.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1, 8, 16

VOIR AUSSI **asf3D**

PROCEDURE **asf3D**

SYNTAXE **asf3D** SeqInStart SeqOutStart Size NbofImages Type

OBJET Calcule le filtre alterné séquentiel de taille *Size* et de base cube-octaédrique ou cubique pour la séquence de *NbofImages* commençant à la zone mémoire n° *SeqInStart*. Le résultat est placé dans la série des mémoires commençant au n° *SeqOutStart*. Si le paramètre *type* vaut 1, le filtre commence par la fermeture **close3D** suivie de **open3D**, sinon c'est l'inverse.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1, 8, 16

VOIR AUSSI **af3D**

PROCEDURE **buildaf3D**

SYNTAXE **buildaf3D** SeqInStart SeqOutStart Size NbofImages Type

OBJET Calcule le filtre alterné par reconstruction de taille *Size* et de base cube-octaédrique ou cubique pour la séquence de *NbofImages* commençant à la zone mémoire n° *SeqInStart*. Le résultat est placé dans la série des mémoires commençant au n° *SeqOutStart*. Si le paramètre *type* vaut 1, le filtre commence par la fermeture **close3D** suivie de **open3D**, sinon c'est l'inverse.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1, 8, 16

EXEMPLE **buildasf3D**

PROCEDURE **buildasf3D**

SYNTAXE **buildasf3D** SeqInStart SeqOutStart Size NbofImages Type

OBJET Calcule le filtre alterné séquentiel par reconstruction de taille *Size* et de base cube-octaédrique ou cubique pour la séquence de *NbofImages* commençant à la zone mémoire n° *SeqInStart*. Le résultat est placé dans la série des mémoires commençant au n° *SeqOutStart*. Si le paramètre *type* vaut 1, le filtre commence par la fermeture **close3D** suivie de **open3D**, sinon c'est l'inverse.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1, 8, 16

VOIR AUSSI **buildaf3D**

PROCEDURE **timdil**

SYNTAXE **timdil** SeqInOutStart Size NbofImages

OBJET Dilatation de la séquence de *NbofImages* commençant à la zone mémoire n° *SeqInStart* , par un segment orthogonal au plan des images et de longueur *Size*. Le résultat est placé dans la même série de mémoires que la séquence d'entrée.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1, 8

VOIR AUSSI **timero**, **timopen**, **timclose**

PROCEDURE **timero**

SYNTAXE **timero** SeqInOutStart Size NbOfImages

OBJET Erosion de la séquence de *NbofImages* commençant à la zone mémoire n° *SeqInStart* , par un segment orthogonal au plan des images et de longueur *Size*. Le résultat est placé dans la même série de mémoires que la séquence d'entrée.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1, 8

VOIR AUSSI **timdil, timopen, timclose**

PROCEDURE **timopen**

SYNTAXE **timopen** SeqInOutStart Size NbOfImages

OBJET Ouverture de la séquence de *NbofImages* commençant à la zone mémoire n° *SeqInStart* , par un segment orthogonal au plan des images et de longueur *Size*. Le résultat est placé dans la même série de mémoires que la séquence d'entrée.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1, 8

VOIR AUSSI **timclose, timero, timdil**

PROCEDURE **timclose**

SYNTAXE **timclose** SeqInOutStart Size NbOfImages

OBJET Fermeture de la séquence de *NbofImages* commençant à la zone mémoire n° *SeqInStart*, par un segment orthogonal au plan des images et de longueur *Size*. Le résultat est placé dans la même série de mémoires que la séquence d'entrée.

MODES *grid* : [1], 0 *edge* : 1, 0 *depth* : 1, 8

EXEMPLE **timopen, timero, timdil**

20. SOURIS

PROCEDURE **fill**

SYNTAXE **fill** imin lev

OBJET L'utilisateur définit en cliquant avec la souris les sommets d'un polygone dont l'intérieur est rempli avec la valeur de gris *lev*. La procédure se termine par un triple clic.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1, 8, 16*

PROCEDURE **dline**

SYNTAXE **dline** imin level

OBJET Trace dans l'image *imin* un (ou plusieurs) segments de valeur de gris *level*, dont les extrémités sont définies avec la souris. La procédure se termine par un triple clic.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1, 8, 16*

PROCEDURE **delobj**

SYNTAXE **delobj** imin

OBJET Dans l'image binaire *imin*, supprime la composante connexe pointée à la souris. Sortie par un triple clic.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1*

PROCEDURE **pointinfo**

SYNTAXE **pointinfo** imin

OBJET Quand l'utilisateur clique sur un point de l'image *imin*, la procédure fournit les coordonnées du point et sa valeur de gris. Ces valeurs sont visualisées dans le bandeau de la fenêtre de visualisation de l'image *imin*. Si *imin* est une image couleur, la valeur affichée correspond au code de la couleur dans la palette. La procédure se termine par un triple clic.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1*

PROCEDURE binpointer

SYNTAXE **binpointer** binin binout type

OBJET Place dans *binout* la particule de *binin* pointée si type vaut 1, ou le point lui-même si type vaut 0. Se termine par un triple clic.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1*

PROCEDURE objinfo

SYNTAXE **objinfo** imin

OBJET Indique dans le bandeau de visualisation de *imin* la surface de la composante connexe pointée à la souris. Se termine par un triple clic.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1*

PROCEDURE brush

SYNTAXE **brush** imin level size

OBJET Ecrit dans in le point défini à la souris, dilaté de taille *size* et avec un niveau de gris *level*. Peut être utilisé soit pour tracer soit pour effacer des particules. Se termine avec un triple clic.

MODES *grid : 1, 0* *edge : 1, 0* *depth : 1*

Table des procédures et fonctions

1. OUTLIS UTILITAIRES

clr	<i>utility.mic</i>	(1-1)
inside	<i>utility.mic</i>	(1-1)
clean	<i>utility.mic</i>	(1-1)
inferior	<i>utility.mic</i>	(1-1)
div2	<i>utility.mic</i>	(1-1)
div	<i>utility.mic</i>	(1-2)
mean	<i>utility.mic</i>	(1-2)
immean	<i>utility.mic</i>	(1-2)
incrust	<i>utility.mic</i>	(1-2)
bintogrey	<i>utility.mic</i>	(1-2)
translate	<i>utility.mic</i>	(1-3)
s4rotate	<i>utility.mic</i>	(1-3)
power	<i>utility.mic</i>	(1-3)
mod	<i>utility.mic</i>	(1-3)
immul	<i>utility.mic</i>	(1-3)
masksup	<i>utility.mic</i>	(1-4)
masksupequal	<i>utility.mic</i>	(1-4)
greymask	<i>utility.mic</i>	(1-4)
abs	<i>utility.mic</i>	(1-4)
greyabs	<i>utility.mic</i>	(1-5)
format	<i>utility.mic</i>	(1-5)
ngbnb	<i>utility.mic</i>	(1-5)
dirtranspose	<i>utility.mic</i>	(1-5)
adjust	<i>utility.mic</i>	(1-5)
getccords	<i>utility.mic</i>	(1-6)
gandb	<i>utility.mic</i>	(1-6)
stop	<i>utility.mic</i>	(1-6)
strech	<i>utility.mic</i>	(1-6)
delta	<i>utility.mic</i>	(1-6)
half	<i>display.mic</i>	(1-7)

2. OUTILS DE VISUALISATION

ds	<i>utility.mic</i>	(2-1)
dscol	<i>display.mic</i>	(2-1)
shadow	<i>display.mic</i>	(2-1)
clrcol	<i>display.mic</i>	(2-2)
col	<i>display.mic</i>	(2-2)
pcol	<i>display.mic</i>	(2-2)
cp	<i>display.mic</i>	(2-2)
cppal	<i>display.mic</i>	(2-3)
refresh	<i>display.mic</i>	(2-3)
profile	<i>display.mic</i>	(2-3)
coldisplay	<i>display.mic</i>	(2-3)

3. EROSIONS, DILATATIONS

dil	<i>erosion.mic</i>	(3-1)
ero	<i>erosion.mic</i>	(3-1)
distance	<i>erosion.mic</i>	(3-1)
dirdil	<i>erosion.mic</i>	(3-2)
direro	<i>erosion.mic</i>	(3-2)
minidil	<i>erosion.mic</i>	(3-3)
miniero	<i>erosion.mic</i>	(3-3)
b1dil	<i>erosion.mic</i>	(3-3)
b2dil	<i>erosion.mic</i>	(3-4)
dbldil	<i>erosion.mic</i>	(3-4)
dblero	<i>erosion.mic</i>	(3-4)
gradient	<i>erosion.mic</i>	(3-4)
sobel	<i>erosion.mic</i>	(3-5)
dilate	<i>erosion.mic</i>	(3-5)
erode	<i>erosion.mic</i>	(3-5)
sq4cont	<i>erosion.mic</i>	(3-5)
cont	<i>erosion.mic</i>	(3-5)
isodil	<i>erosion2.mic</i>	(3-6)
isoero	<i>erosion2.mic</i>	(3-6)
isodist	<i>erosion2.mic</i>	(3-6)
conedil	<i>erosion2.mic</i>	(3-6)
cyldil	<i>erosion2.mic</i>	(3-7)
crossdil	<i>erosion2.mic</i>	(3-7)
rhombodil	<i>erosion2.mic</i>	(3-7)
rhomboero	<i>erosion2.mic</i>	(3-8)
bero	<i>erosion2.mic</i>	(3-8)
diamdil	<i>erosion2.mic</i>	(3-8)
diamero	<i>erosion2.mic</i>	(3-8)
sh1dil	<i>erosion2.mic</i>	(3-9)
sh2dil	<i>erosion2.mic</i>	(3-9)
sh1ero	<i>erosion2.mic</i>	(3-9)
sh2ero	<i>erosion2.mic</i>	(3-9)
shdil	<i>erosion2.mic</i>	(3-10)
shero	<i>erosion2.mic</i>	(3-10)
ringdil1	<i>erosion2.mic</i>	(3-10)
ringdil2	<i>erosion2.mic</i>	(3-10)

4. OPERATEURS DE RANG

rank	<i>rank.mic</i>	(4-1)
median	<i>rank.mic</i>	(4-1)
binsegmi	<i>rank.mic</i>	(4-1)

5. CONVOLUTIONS

hgauss1	<i>convol.mic</i>	(5-1)
vgauss1	<i>convol.mic</i>	(5-1)
gauss1	<i>convol.mic</i>	(5-1)
gauss	<i>convol.mic</i>	(5-2)

6. OVERTURES, FERMETURES

open	<i>opening.mic</i>	(6-1)
close	<i>opening.mic</i>	(6-1)
diropen	<i>opening.mic</i>	(6-1)
dirclose	<i>opening.mic</i>	(6-2)
lineopen	<i>opening.mic</i>	(6-2)
lineclose	<i>opening.mic</i>	(6-2)
openth	<i>opening.mic</i>	(6-2)
closeth	<i>opening.mic</i>	(6-3)
lineopenth	<i>opening.mic</i>	(6-3)
linecloseth	<i>opening.mic</i>	(6-3)
pregrad	<i>opening.mic</i>	(6-3)
regrad	<i>opening.mic</i>	(6-4)
miniopen	<i>opening2.mic</i>	(6-5)
miniclose	<i>opening2.mic</i>	(6-5)
isopen	<i>opening2.mic</i>	(6-5)
isoclose	<i>opening2.mic</i>	(6-5)
inlopen	<i>opening2.mic</i>	(6-6)
supclose	<i>opening2.mic</i>	(6-6)
stopen	<i>opening2.mic</i>	(6-6)

7. GEODESIE ET CONNEXITE

gdsdil	<i>geodesy.mic</i>	(7-1)
gdsero	<i>geodesy.mic</i>	(7-1)
build	<i>geodesy.mic</i>	(7-1)
buildopen	<i>geodesy.mic</i>	(7-1)
buildclose	<i>geodesy.mic</i>	(7-2)
areaopen	<i>geodesy.mic</i>	(7-2)
ringbuild	<i>geodesy.mic</i>	(7-2)
gdsdist	<i>geodesy.mic</i>	(7-2)
recons	<i>geodesy.mic</i>	(7-3)
levelling	<i>geodesy.mic</i>	(7-3)

8. APPLICATIONS DE LA GEODESIE

edgeoff	<i>geod_use.mic</i>	(8-1)
fgrain	<i>geod_use.mic</i>	(8-1)
border	<i>geod_use.mic</i>	(8-1)
clohole	<i>geod_use.mic</i>	(8-1)

maxima	<i>geod_use.mic</i>	(8-1)
minima	<i>geod_use.mic</i>	(8-2)
extmaxima	<i>geod_use.mic</i>	(8-2)
extminima	<i>geod_use.mic</i>	(8-2)
swamping	<i>geod_use.mic</i>	(8-2)
extrema	<i>geod_use.mic</i>	(8-2)
grainclose	<i>geod_use.mic</i>	(8-3)
indivaf	<i>geod_use.mic</i>	(8-3)
dynamics	<i>geod_use.mic</i>	(8-3)

9. FILTRES MORPHOLOGIQUES

af	<i>filters.mic</i>	(9-1)
fullasf	<i>filters.mic</i>	(9-1)
lineaf	<i>filters.mic</i>	(9-1)
asf	<i>filters.mic</i>	(9-2)
lineasf	<i>filters.mic</i>	(9-2)
automed	<i>filters.mic</i>	(9-2)
centre	<i>filters.mic</i>	(9-2)
contrast	<i>filters.mic</i>	(9-3)
contrasth	<i>filters.mic</i>	(9-3)
minifilt	<i>filters2.mic</i>	(9-4)
isaf	<i>filters2.mic</i>	(9-4)
isasf	<i>filters2.mic</i>	(9-4)
buildaf	<i>filters2.mic</i>	(9-4)
buildasf	<i>filters2.mic</i>	(9-5)
minibuildaf	<i>geodesy.mic</i>	(9-5)
closoropen	<i>filters2.mic</i>	(9-5)
binmiddle	<i>filters2.mic</i>	(9-5)
grmiddle	<i>filters2.mic</i>	(9-6)

10 BOULES MAXIMALES ET SQUELETTE

binopenskel	<i>skeleton.mic</i>	(10-1)
binultim	<i>skeleton.mic</i>	(10-1)
centroid	<i>skeleton.mic</i>	(10-1)
condbis	<i>skeleton.mic</i>	(10-1)

11. AMINCIÈSSEMENTS, EPAISSISSEMENTS

thickturn	<i>thinning.mic</i>	(11-2)
gdsthickturn	<i>thinning.mic</i>	(11-2)
thick	<i>thinning.mic</i>	(11-2)
gdsthick	<i>thinning.mic</i>	(11-2)
thinturn	<i>thinning.mic</i>	(11-2)
gdsthinturn	<i>thinning.mic</i>	(11-2)
thin	<i>thinning.mic</i>	(11-3)
gdsthin	<i>thinning.mic</i>	(11-3)

Lthick	<i>thinning.mic</i>	(11-3)
Mthick	<i>thinning.mic</i>	(11-3)
Dthick	<i>thinning.mic</i>	(11-4)
Dthin	<i>thinning.mic</i>	(11-4)
Lthin	<i>thinning.mic</i>	(11-5)
Mthin	<i>thinning.mic</i>	(11-5)
endpoints	<i>thinning.mic</i>	(11-5)
mulpoints	<i>thinning.mic</i>	(11-5)
delsp	<i>thinning.mic</i>	(11-6)
gdscentre	<i>thinning.mic</i>	(11-6)
greyseero	<i>thinning.mic</i>	(11-7)
greysedil	<i>thinning.mic</i>	(11-7)
greythinstep	<i>thinning.mic</i>	(11-7)
greythickstep	<i>thinning.mic</i>	(11-7)
dirgradient	<i>thinning.mic</i>	(11-8)
vectgrad0	<i>thinning.mic</i>	(11-8)
gradvect	<i>thinning.mic</i>	(11-8)
mainthin	<i>thinning.mic</i>	(11-8)
greyhmt	<i>thinning.mic</i>	(11-9)
greythickturn	<i>thinning.mic</i>	(11-9)
greythick	<i>thinning.mic</i>	(11-9)

12. LIGNE DE PARTAGE DES EAUX

clip	<i>wshed.mic</i>	(12-1)
skiz	<i>wshed.mic</i>	(12-1)
gdskez	<i>wshed.mic</i>	(12-1)
threshwshed	<i>wshed.mic</i>	(12-1)
mwsheed	<i>wshed.mic</i>	(12-2)
wshed	<i>wshed.mic</i>	(12-2)

13. SEGMENTATION

mgradwshed	<i>segment.mic</i>	(13-1)
gradwshed	<i>segment.mic</i>	(13-1)
shapeseq	<i>segment.mic</i>	(13-1)
smoothcnc	<i>segment.mic</i>	(13-1)
jumpcnc	<i>segment.mic</i>	(13-2)
mosaic	<i>segment.mic</i>	(13-2)
wfall	<i>segment.mic</i>	(13-3)
kheops	<i>segment.mic</i>	(13-3)
jump	<i>segment.mic</i>	(13-3)

14. MESURES

diameter	<i>stereo.mic</i>	(14-2)
digerim	<i>stereo.mic</i>	(14-2)
perim	<i>stereo.mic</i>	(14-3)

cnumber	<i>stereo.mic</i>	(14-3)
binhverret	<i>stereo.mic</i>	(14-3)
binhferret	<i>stereo.mic</i>	(14-3)
var0	<i>statist.mic</i>	(14-4)
var1	<i>statist.mic</i>	(14-4)
var2	<i>statist.mic</i>	(14-4)
rug	<i>statist.mic</i>	(14-4)
mse	<i>statist.mic</i>	(14-5)
bincount	<i>stereo.mic</i>	(14-5)
floatcount	<i>stereo.mic</i>	(14-5)
flatzone	<i>stereo.mic</i>	(14-5)

15. COURBES

bincov	<i>curves.mic</i>	(15-1)
covar	<i>curves.mic</i>	(15-1)
vario1	<i>curves.mic</i>	(15-1)
vario2	<i>curves.mic</i>	(15-2)
modcont	<i>curves.mic</i>	(15-2)
isogranul	<i>curves.mic</i>	(15-3)
granul	<i>curves.mic</i>	(15-3)
pvonl	<i>curves.mic</i>	(15-3)
p1vonl	<i>curves.mic</i>	(15-4)
p2vonl	<i>curves.mic</i>	(15-4)

16. SIMULATIONS ALEATOIRES

point	<i>simul.mic</i>	(16-1)
points	<i>simul.mic</i>	(16-1)
regpoints	<i>simul.mic</i>	(16-1)
isobool	<i>simul.mic</i>	(16-1)
isobool2	<i>simul.mic</i>	(16-2)
eldil	<i>simul.mic</i>	(16-2)
dropdil	<i>simul.mic</i>	(16-2)
elbool	<i>simul.mic</i>	(16-2)
dropbool	<i>simul.mic</i>	(16-2)
tribool	<i>simul.mic</i>	(16-2)
hard	<i>simul.mic</i>	(16-2)
rose	<i>simul.mic</i>	(16-3)
disjoint	<i>simul.mic</i>	(16-3)
flake	<i>simul.mic</i>	(16-3)
conebool	<i>simul.mic</i>	(16-4)
conebool2	<i>simul.mic</i>	(16-4)
rocky	<i>simul.mic</i>	(16-4)
lines	<i>simul.mic</i>	(16-5)
diags	<i>simul.mic</i>	(16-5)
steps	<i>simul.mic</i>	(16-5)
nestlines	<i>simul.mic</i>	(16-5)

17. GRAPHERS

Gdil	<i>graph.mic</i>	(17-1)
Gero	<i>graph.mic</i>	(17-1)
Gopen	<i>graph.mic</i>	(17-1)
Gclose	<i>graph.mic</i>	(17-2)
setborder	<i>graph.mic</i>	(17-2)
Gbuild	<i>graph.mic</i>	(17-2)
label	<i>graph.mic</i>	(17-2)

18. UTILITAIRES 3D

Seqload	<i>3dutil.mic</i>	(18-1)
Seqsave	<i>3dutil.mic</i>	(18-2)
Seqclr	<i>3dutil.mic</i>	(18-2)
Seqset	<i>3dutil.mic</i>	(18-2)
Seqcopy	<i>3dutil.mic</i>	(18-3)
Seqinv	<i>3dutil.mic</i>	(18-3)
Seqadd	<i>3dutil.mic</i>	(18-3)
Seqcadd	<i>3dutil.mic</i>	(18-4)
Seqseqadd	<i>3dutil.mic</i>	(18-4)
Seqcsub	<i>3dutil.mic</i>	(18-4)
Seqseqsub	<i>3dutil.mic</i>	(18-5)
Seqcmul	<i>3dutil.mic</i>	(18-5)
Seqmean	<i>3dutil.mic</i>	(18-6)
Seqseqmean	<i>3dutil.mic</i>	(18-6)
Seqdiff	<i>3dutil.mic</i>	(18-6)
Seqvolume	<i>3dutil.mic</i>	(18-7)
Seqinf	<i>3dutil.mic</i>	(18-7)
Seqseqinf	<i>3dutil.mic</i>	(18-8)
Seqsup	<i>3dutil.mic</i>	(18-8)
Seqseqsup	<i>3dutil.mic</i>	(18-8)
Seqbuild	<i>3dutil.mic</i>	(18-9)
Seqthresh	<i>3dutil.mic</i>	(18-9)
Seqgrad	<i>3dutil.mic</i>	(18-10)
Seqhist	<i>3dutil.mic</i>	(18-10)
Scroll	<i>3dutil.mic</i>	(18-10)
Scroll2	<i>3dutil.mic</i>	(18-11)
Scroll3	<i>3dutil.mic</i>	(18-11)
Visu	<i>3dutil.mic</i>	(18-11)

19. PROCEDURES 3D

ero3D	<i>3dproces.mic</i>	(19-1)
dil3D	<i>3dproces.mic</i>	(19-1)
open3D	<i>3dproces.mic</i>	(19-2)
close3D	<i>3dproces.mic</i>	(19-2)
gradient3D	<i>3dproces.mic</i>	(19-2)

build3D	<i>3dproces.mic</i>	(19-3)
buildopen3D	<i>3dproces.mic</i>	(19-3)
buildclose3D	<i>3dproces.mic</i>	(19-3)
openth3D	<i>3dproces.mic</i>	(19-4)
closeth3D	<i>3dproces.mic</i>	(19-4)
maxima3D	<i>3dproces.mic</i>	(19-5)
minima3D	<i>3dproces.mic</i>	(19-5)
af3D	<i>3dproces.mic</i>	(19-5)
asf3D	<i>3dproces.mic</i>	(19-5)
buildaf3D	<i>3dproces.mic</i>	(19-6)
buildasf3D	<i>3dproces.mic</i>	(19-6)
timdil	<i>3dproces.mic</i>	(19-6)
timero	<i>3dproces.mic</i>	(19-7)
timopen	<i>3dproces.mic</i>	(19-7)
timclose	<i>3dproces.mic</i>	(19-7)

20. SOURIS

fill	<i>mouse.mic</i>	(20-1)
dline	<i>mouse.mic</i>	(20-1)
delobj	<i>mouse.mic</i>	(20-1)
pointinfo	<i>mouse.mic</i>	(20-1)
binpointer	<i>mouse.mic</i>	(20-2)
objinfo	<i>mouse.mic</i>	(20-2)
brush	<i>mouse.mic</i>	(20-2)

LISTE ALPHABETIQUE DES MOTS COMPOSES

abs	<i>utility.mic</i>	(1-4)
adjust	<i>utility.mic</i>	(1-5)
af	<i>filters.mic</i>	(9-1)
af3D	<i>3dproces.mic</i>	(19-5)
areaopen	<i>geodesy.mic</i>	(7-2)
asf	<i>filters.mic</i>	(9-2)
asf3D	<i>3dproces.mic</i>	(19-5)
automed	<i>filters.mic</i>	(9-2)
b1dil	<i>erosion.mic</i>	(3-3)
b2dil	<i>erosion.mic</i>	(3-4)
bero	<i>erosion2.mic</i>	(3-8)
bincount	<i>stereo.mic</i>	(14-5)
bincov	<i>curves.mic</i>	(15-1)
binhferret	<i>stereo.mic</i>	(14-3)
binhverret	<i>stereo.mic</i>	(14-3)
binmiddle	<i>filters2.mic</i>	(9-5)
binopenskel	<i>skeleton.mic</i>	(10-1)
binpointer	<i>mouse.mic</i>	(20-2)
binsegmi	<i>rank.mic</i>	(4-1)
bintogrey	<i>utility.mic</i>	(1-2)
binultim	<i>skeleton.mic</i>	(10-1)
border	<i>geod_use.mic</i>	(8-1)
brush	<i>mouse.mic</i>	(20-2)
build	<i>geodesy.mic</i>	(7-1)
build3D	<i>3dproces.mic</i>	(19-3)
buildaf	<i>filters2.mic</i>	(9-4)
buildaf3D	<i>3dproces.mic</i>	(19-6)
buildasf	<i>filters2.mic</i>	(9-5)
buildasf3D	<i>3dproces.mic</i>	(19-6)
buildclose	<i>geodesy.mic</i>	(7-2)
buildclose3D	<i>3dproces.mic</i>	(19-3)
buildopen	<i>geodesy.mic</i>	(7-1)
buildopen3D	<i>3dproces.mic</i>	(19-3)
centre	<i>filters.mic</i>	(9-2)
centroid	<i>skeleton.mic</i>	(10-1)
clean	<i>utility.mic</i>	(1-1)
clip	<i>wshed.mic</i>	(12-1)
clohole	<i>geod_use.mic</i>	(8-1)
close	<i>opening.mic</i>	(6-1)
close3D	<i>3dproces.mic</i>	(19-2)
closeth	<i>opening.mic</i>	(6-3)
closeth3D	<i>3dproces.mic</i>	(19-4)
closoropen	<i>filters2.mic</i>	(9-5)
clr	<i>utility.mic</i>	(1-1)
clrcol	<i>display.mic</i>	(2-2)
cnumber	<i>stereo.mic</i>	(14-3)

col	<i>display.mic</i>	(2-2)
coldisplay	<i>display.mic</i>	(2-3)
condbis	<i>skeleton.mic</i>	(10-1)
conebool	<i>simul.mic</i>	(16-4)
conebool2	<i>simul.mic</i>	(16-4)
conedil	<i>erosion2.mic</i>	(3-6)
cont	<i>erosion.mic</i>	(3-5)
contrast	<i>filters.mic</i>	(9-3)
contrasth	<i>filters.mic</i>	(9-3)
covar	<i>curves.mic</i>	(15-1)
cp	<i>display.mic</i>	(2-2)
cppal	<i>display.mic</i>	(2-3)
crossdil	<i>erosion2.mic</i>	(3-7)
cyldil	<i>erosion2.mic</i>	(3-7)
dbldil	<i>erosion.mic</i>	(3-4)
dblero	<i>erosion.mic</i>	(3-4)
delobj	<i>mouse.mic</i>	(20-1)
delsp	<i>thinning.mic</i>	(11-6)
delta	<i>utility.mic</i>	(1-6)
diags	<i>simul.mic</i>	(16-5)
diamdil	<i>erosion2.mic</i>	(3-8)
diamero	<i>erosion2.mic</i>	(3-8)
diameter	<i>stereo.mic</i>	(14-2)
digperim	<i>stereo.mic</i>	(14-2)
dil	<i>erosion.mic</i>	(3-1)
dil3D	<i>3dproces.mic</i>	(19-1)
dilate	<i>erosion.mic</i>	(3-5)
dirclose	<i>opening.mic</i>	(6-2)
dirdil	<i>erosion.mic</i>	(3-2)
direro	<i>erosion.mic</i>	(3-2)
dirgradient	<i>thinning.mic</i>	(11-8)
diropen	<i>opening.mic</i>	(6-1)
dirtranspose	<i>utility.mic</i>	(1-5)
disjoint	<i>simul.mic</i>	(16-3)
distance	<i>erosion.mic</i>	(3-1)
div	<i>utility.mic</i>	(1-2)
div2	<i>utility.mic</i>	(1-1)
dline	<i>mouse.mic</i>	(20-1)
dropbool	<i>simul.mic</i>	(16-2)
dropdil	<i>simul.mic</i>	(16-2)
ds	<i>utility.mic</i>	(2-1)
dscol	<i>display.mic</i>	(2-1)
Dthick	<i>thinning.mic</i>	(11-4)
Dthin	<i>thinning.mic</i>	(11-4)
dynamics	<i>geod_use.mic</i>	(8-3)
edgeoff	<i>geod_use.mic</i>	(8-1)
elbool	<i>simul.mic</i>	(16-2)
eldil	<i>simul.mic</i>	(16-2)

endpoints	<i>thinning.mic</i>	(11-5)
ero	<i>erosion.mic</i>	(3-1)
ero3D	<i>3dproces.mic</i>	(19-1)
erode	<i>erosion.mic</i>	(3-5)
extmaxima	<i>geod_use.mic</i>	(8-2)
extminima	<i>geod_use.mic</i>	(8-2)
extrema	<i>geod_use.mic</i>	(8-2)
fgrain	<i>geod_use.mic</i>	(8-1)
fill	<i>mouse.mic</i>	(20-1)
flake	<i>simul.mic</i>	(16-3)
flatzone	<i>stereo.mic</i>	(14-5)
floatcount	<i>stereo.mic</i>	(14-5)
format	<i>utility.mic</i>	(1-5)
fullasf	<i>filters.mic</i>	(9-1)
gandb	<i>utility.mic</i>	(1-6)
gauss	<i>convol.mic</i>	(5-2)
gauss1	<i>convol.mic</i>	(5-1)
Gbuild	<i>graph.mic</i>	(17-2)
Gclose	<i>graph.mic</i>	(17-2)
Gdil	<i>graph.mic</i>	(17-1)
gdscentre	<i>thinning.mic</i>	(11-6)
gdsdil	<i>geodesy.mic</i>	(7-1)
gdsdist	<i>geodesy.mic</i>	(7-2)
gdsero	<i>geodesy.mic</i>	(7-1)
gdskiz	<i>wshed.mic</i>	(12-1)
gdsthick	<i>thinning.mic</i>	(11-2)
gdsthickturn	<i>thinning.mic</i>	(11-2)
gdsthin	<i>thinning.mic</i>	(11-3)
gdsthinturn	<i>thinning.mic</i>	(11-2)
Gero	<i>graph.mic</i>	(17-1)
getccords	<i>utility.mic</i>	(1-6)
Gopen	<i>graph.mic</i>	(17-1)
gradient	<i>erosion.mic</i>	(3-4)
gradient3D	<i>3dproces.mic</i>	(19-2)
gradvect	<i>thinning.mic</i>	(11-8)
gradwshed	<i>segment.mic</i>	(13-1)
grainclose	<i>geod_use.mic</i>	(8-3)
granul	<i>curves.mic</i>	(15-3)
greyabs	<i>utility.mic</i>	(1-5)
greyhmt	<i>thinning.mic</i>	(11-9)
greymask	<i>utility.mic</i>	(1-4)
greysedil	<i>thinning.mic</i>	(11-7)
greyseero	<i>thinning.mic</i>	(11-7)
greythick	<i>thinning.mic</i>	(11-9)
greythickstep	<i>thinning.mic</i>	(11-7)
greythickturn	<i>thinning.mic</i>	(11-9)
greythinstep	<i>thinning.mic</i>	(11-7)
grmiddle	<i>filters2.mic</i>	(9-6)

half	<i>display.mic</i>	(1-7)
hard	<i>simul.mic</i>	(16-2)
hgauss1	<i>convol.mic</i>	(5-1)
immean	<i>utility.mic</i>	(1-2)
immul	<i>utility.mic</i>	(1-3)
incrust	<i>utility.mic</i>	(1-2)
indivaf	<i>geod_use.mic</i>	(8-3)
inferior	<i>utility.mic</i>	(1-1)
inlopen	<i>opening2.mic</i>	(6-6)
inside	<i>utility.mic</i>	(1-1)
isaf	<i>filters2.mic</i>	(9-4)
isaf	<i>filters2.mic</i>	(9-4)
isobool	<i>simul.mic</i>	(16-1)
isobool2	<i>simul.mic</i>	(16-2)
isoclose	<i>opening2.mic</i>	(6-5)
isodil	<i>erosion2.mic</i>	(3-6)
isodist	<i>erosion2.mic</i>	(3-6)
isoero	<i>erosion2.mic</i>	(3-6)
isogranul	<i>curves.mic</i>	(15-3)
isopen	<i>opening2.mic</i>	(6-5)
jump	<i>segment.mic</i>	(13-3)
jumpcnc	<i>segment.mic</i>	(13-2)
kheops	<i>segment.mic</i>	(13-3)
label	<i>graph.mic</i>	(17-2)
levelling	<i>geodesy.mic</i>	(7-3)
lineaf	<i>filters.mic</i>	(9-1)
lineasf	<i>filters.mic</i>	(9-2)
lineclose	<i>opening.mic</i>	(6-2)
linecloseth	<i>opening.mic</i>	(6-3)
lineopen	<i>opening.mic</i>	(6-2)
lineopenth	<i>opening.mic</i>	(6-3)
lines	<i>simul.mic</i>	(16-5)
Lthick	<i>thinning.mic</i>	(11-3)
Lthin	<i>thinning.mic</i>	(11-5)
mainthin	<i>thinning.mic</i>	(11-8)
masksup	<i>utility.mic</i>	(1-4)
masksupequal	<i>utility.mic</i>	(1-4)
maxima	<i>geod_use.mic</i>	(8-1)
maxima3D	<i>3dproces.mic</i>	(19-5)
mean	<i>utility.mic</i>	(1-2)
median	<i>rank.mic</i>	(4-1)
mgradwshed	<i>segment.mic</i>	(13-1)
minibuildaf	<i>geodesy.mic</i>	(9-5)
miniclose	<i>opening2.mic</i>	(6-5)
minidil	<i>erosion.mic</i>	(3-3)
miniero	<i>erosion.mic</i>	(3-3)
minifilt	<i>filters2.mic</i>	(9-4)
minima	<i>geod_use.mic</i>	(8-2)

minima3D	<i>3dproces.mic</i>	(19-5)
miniopen	<i>opening2.mic</i>	(6-5)
mod	<i>utility.mic</i>	(1-3)
modcont	<i>curves.mic</i>	(15-2)
mosaic	<i>segment.mic</i>	(13-2)
mse	<i>statist.mic</i>	(14-5)
Mthick	<i>thinning.mic</i>	(11-3)
Mthin	<i>thinning.mic</i>	(11-5)
mulpoints	<i>thinning.mic</i>	(11-5)
mwshed	<i>wshed.mic</i>	(12-2)
nestlines	<i>simul.mic</i>	(16-5)
ngbnb	<i>utility.mic</i>	(1-5)
objinfo	<i>mouse.mic</i>	(20-2)
open	<i>opening.mic</i>	(6-1)
open3D	<i>3dproces.mic</i>	(19-2)
openth	<i>opening.mic</i>	(6-2)
openth3D	<i>3dproces.mic</i>	(19-4)
p1vonl	<i>curves.mic</i>	(15-4)
p2vonl	<i>curves.mic</i>	(15-4)
perim	<i>stereo.mic</i>	(14-3)
point	<i>simul.mic</i>	(16-1)
pointinfo	<i>mouse.mic</i>	(20-1)
points	<i>simul.mic</i>	(16-1)
power	<i>utility.mic</i>	(1-3)
pregrad	<i>opening.mic</i>	(6-3)
profile	<i>display.mic</i>	(2-3)
pscol	<i>display.mic</i>	(2-2)
pvonl	<i>curves.mic</i>	(15-3)
rank	<i>rank.mic</i>	(4-1)
recons	<i>geodesy.mic</i>	(7-3)
refresh	<i>display.mic</i>	(2-3)
regpoints	<i>simul.mic</i>	(16-1)
regrad	<i>opening.mic</i>	(6-4)
rhombodil	<i>erosion2.mic</i>	(3-7)
rhomboero	<i>erosion2.mic</i>	(3-8)
ringbuild	<i>geodesy.mic</i>	(7-2)
ringdil1	<i>erosion2.mic</i>	(3-10)
ringdil2	<i>erosion2.mic</i>	(3-10)
rocky	<i>simul.mic</i>	(16-4)
rose	<i>simul.mic</i>	(16-3)
rug	<i>statist.mic</i>	(14-4)
s4rotate	<i>utility.mic</i>	(1-3)
Scroll	<i>3dutil.mic</i>	(18-10)
Scroll2	<i>3dutil.mic</i>	(18-11)
Scroll3	<i>3dutil.mic</i>	(18-11)
Seqadd	<i>3dutil.mic</i>	(18-3)
Seqbuild	<i>3dutil.mic</i>	(18-9)
Seqcadd	<i>3dutil.mic</i>	(18-4)

Seqclr	<i>3dutil.mic</i>	(18-2)
Seqcmul	<i>3dutil.mic</i>	(18-5)
Seqcopy	<i>3dutil.mic</i>	(18-3)
Seqcsub	<i>3dutil.mic</i>	(18-4)
Seqdiff	<i>3dutil.mic</i>	(18-6)
Seqgrad	<i>3dutil.mic</i>	(18-10)
Seqhist	<i>3dutil.mic</i>	(18-10)
Seqinf	<i>3dutil.mic</i>	(18-7)
Seqinv	<i>3dutil.mic</i>	(18-3)
Seqload	<i>3dutil.mic</i>	(18-1)
Seqmean	<i>3dutil.mic</i>	(18-6)
Seqsave	<i>3dutil.mic</i>	(18-2)
Seqseqadd	<i>3dutil.mic</i>	(18-4)
Seqseqinf	<i>3dutil.mic</i>	(18-8)
Seqseqmean	<i>3dutil.mic</i>	(18-6)
Seqseqsub	<i>3dutil.mic</i>	(18-5)
Seqseqsup	<i>3dutil.mic</i>	(18-8)
Seqset	<i>3dutil.mic</i>	(18-2)
Seqsup	<i>3dutil.mic</i>	(18-8)
Seqthresh	<i>3dutil.mic</i>	(18-9)
Seqvolume	<i>3dutil.mic</i>	(18-7)
setborder	<i>graph.mic</i>	(17-2)
sh1dil	<i>erosion2.mic</i>	(3-9)
sh1ero	<i>erosion2.mic</i>	(3-9)
sh2dil	<i>erosion2.mic</i>	(3-9)
sh2ero	<i>erosion2.mic</i>	(3-9)
shadow	<i>display.mic</i>	(2-1)
shapeseq	<i>segment.mic</i>	(13-1)
shdil	<i>erosion2.mic</i>	(3-10)
shero	<i>erosion2.mic</i>	(3-10)
skiz	<i>wshed.mic</i>	(12-1)
smoothcnc	<i>segment.mic</i>	(13-1)
sobel	<i>erosion.mic</i>	(3-5)
sq4cont	<i>erosion.mic</i>	(3-5)
steps	<i>simul.mic</i>	(16-5)
stop	<i>utility.mic</i>	(1-6)
stopen	<i>opening2.mic</i>	(6-6)
strech	<i>utility.mic</i>	(1-6)
supclose	<i>opening2.mic</i>	(6-6)
swamping	<i>geod_use.mic</i>	(8-2)
thick	<i>thinning.mic</i>	(11-2)
thickturn	<i>thinning.mic</i>	(11-2)
thin	<i>thinning.mic</i>	(11-3)
thinturn	<i>thinning.mic</i>	(11-2)
threshwshed	<i>wshed.mic</i>	(12-1)
timclose	<i>3dproces.mic</i>	(19-7)
timdil	<i>3dproces.mic</i>	(19-6)
timero	<i>3dproces.mic</i>	(19-7)

timopen	<i>3dproces.mic</i>	(19-7)
translate	<i>utility.mic</i>	(1-3)
tribool	<i>simul.mic</i>	(16-2)
var0	<i>statist.mic</i>	(14-4)
var1	<i>statist.mic</i>	(14-4)
var2	<i>statist.mic</i>	(14-4)
vario1	<i>curves.mic</i>	(15-1)
vario2	<i>curves.mic</i>	(15-2)
vectgrad0	<i>thinning.mic</i>	(11-8)
vgauss1	<i>convol.mic</i>	(5-1)
Visu	<i>3dutil.mic</i>	(18-11)
wfall	<i>segment.mic</i>	(13-3)
wshed	<i>wshed.mic</i>	(12-2)