

NOTICE

**JEAN-FRANCOIS PARROT
et HIND TAUD**

LOGICIEL ADONIS

Automatic Detection

Of Numerical Image Structures

Février 1991

**LABORATOIRE DE TELEDETECTION
UNIVERSITE PIERRE ET MARIE CURIE**

**MANUEL D'UTILISATION DES PROGRAMMES DU LOGICIEL
ADONIS
(Automatic Detection Of Numerical Image Structures)**

Jean-Francois PARROT

et

Hind TAUD

AVANT-PROPOS

Le logiciel ADONIS (Automatic Detection Of Numerical Image Structures) concerne la détection automatique des structures sub-circulaires sur les images satellitaires. Il a été mis au point et exposé dans les publications de TAUD (1989) et TAUD et PARROT (1990a, 1990b). Il conviendra donc de se reporter, notamment à cette dernière publication, pour avoir plus de détails sur l'ensemble des algorithmes.

Dans ses grandes lignes, la méthode comprend deux volets essentiels :

- (a) mise en évidence des contours sur l'image satellitaire;
- (b) détection des segments sub-circulaires.

Par ailleurs, nombre de programmes assurent en aval de la détection proprement dite, l'utilisation des variables discriminantes de "contour" et de "surface" définies au cours du traitement, et l'exploitation statistique de ces informations.

L'ensemble des programmes décrits dans cette notice rend en partie obsolète le chapitre IX du logiciel TAPIS (PARROT 1990) relatif à la "détection des structures circulaires", dans la mesure où cet ensemble a été repensé et redéfini aux vues des nombreux ajouts apportés à la méthode.

PARROT J.-F. (1990).— Logiciel TAPIS (Traitements Apollo Pour Images Satellitaires). *Lab. Télédétection UPMC Paris VI*, 100p.

TAUD H. (1989).— Détection des structures circulaires sur image numérique : recherche non supervisée des structures géologiques sur image satellitaire. Th. 3ème Cycle, Univ. Mohammed V, Rabat, 137 p.

TAUD H. and PARROT J.-F. (1990).— Detection of circular structures on satellite images. *Int. J. Remote Sensing*, (in press).

TAUD H. et PARROT J.-F. (1990).— Analyse numérique des formes sub-circulaires sur les images satellitaires. *2ème Journées ORSTOM Télédétection* (sous presse).

SOMMAIRE

I.- Organisation et format des fichiers images

- (a) organisation générale
- (b) format image
- (c) descriptif
- (d) format des tableaux répertoriant les contours

II.- Traitements

II.a- extraction de contours significatifs

- (1) nature des données à traiter
- (2) extraction des contours de type (a)
- (3) extraction des contours de type (b)

COCON	: Connection de contours
COCON_VIS	: idem + visualisation
DEST_ANG *	: Elimination des pixels d'angle droit
DEST_CADRE *	: Elimination du cadre
EQUIPOP	: Segmentation de l'image par équidensité
ERODIL	: Erosion et Dilatation
FILTRES	: Divers types de Filtres
INDICES	: Indices de végétation, de brillance et de couleur
LIMITE *	: Limites des objets isolés
PARPON	: Partition pondérée en tranches de pourcentage
PARSIFAL	: Partition + Filtrage + Limite
PARSIFAL2	: Partition + Filtrage + Limite (n images obtenues)
REGIME	; squelettisation itérative
REGIME_VIS	: idem + visualisation

II.b- détection des segments sub-circulaires

- (1) individualisation des contours
- (2) décomposition des courbes en éléments sub-circulaires.
- (3) application de la Transformée de Hough
- (4) calcul de la position exacte du centre

ADONIS	: détection des structures sub-circulaires
ADONIS_VIS	: idem + visualisation

II.c- résultats

- (1) définition des variables de "contour"
- (2) notion d'occurrence
- (3) rapport des variables aux formes
- (4) les variables de "surface"

ASCOR	: Association des tableaux .COR
ASTICOT	: Assemblage des tableaux .RES et .TAB
CAPNOR	: Calcul de la pente de la normale
CONCERT_VIS	: Contenu des cercles détectés + visualisation
CONCORDE	; Calcul de l'occurrence
ECLAT_STAT	: Extraction des classes après analyse statistique
HISBID_COR	: Histogramme bi-dimensionnel DIR/PNC
HIST_COR	: Histogrammes des variables
MIN_MAX	: Minimum, maximum et moyenne des classes
ROSACE	: rosaces des directions
TRICOT	; Tri des courbes

II.d- Analyses statistiques

PRESTA : transfert vers STATITCF

II.e- Transfert Apollo Imprimante Laser

PRELAS_A ; Préparation Impression Laser

III.- Fiches programme

IV.- Modules utilitaires

Nota : les programmes dont les noms sont suivis d'une astérisque sont extraits du logiciel TAPIS.

I. FORMAT DES FICHIERS IMAGES ET DES TABLEAUX ASSOCIES

I.A. Organisation générale

Les fichiers images doivent être systématiquement placés sous le répertoire "user" de l'un des noeuds du réseau (p. ex. node_df91). Un sous-répertoire indique quel est le nom générique où sont consignées toutes les images et les fichiers de l'utilisateur, par exemple :

`//node_df91/user/azrou`

Le sous-répertoire `//node_df91/user/azrou` comprend lui-même une branche "images" où sont placées toutes les images à traiter :

`//node_df91/user/azrou/images`

Ce sous-répertoire comprend également une branche "tables" où se placeront les tableaux créés et exploités par l'ensemble de programmes destiné à la recherche et à la détection des structures circulaires.

par exemple, l'image "canal3" sera située dans la branche "images" du sous-répertoire "azrou" :

`//node_df91/user/azrou/images/canal3`

et le tableau "resultat" sera situé dans la branche "tables" du même sous-répertoire :

`//node_df91/user/azrou/tables/resultat`

En fait, en ce qui concerne les images proprement dites, la branche `//node_df91/user/azrou/images/canal3` est découpée en deux sous-branches, la première comprenant l'image elle-même, la seconde le descriptif de cette image (cf. plus loin).

exemple :

`//node_df91/user/azrou/images/canal3/image` contient l'image

`//node_df91/user/azrou/images/canal3/descriptif` contient le descriptif de cette image.

Cette branche peut également contenir des tableaux, des bitmaps, etc.. correspondant à l'image traitée.

exemples :

`//node_df91/user/azrou/images/canal3/tab1_histo`

`//node_df91/user/azrou/images/canal3/bitmap`

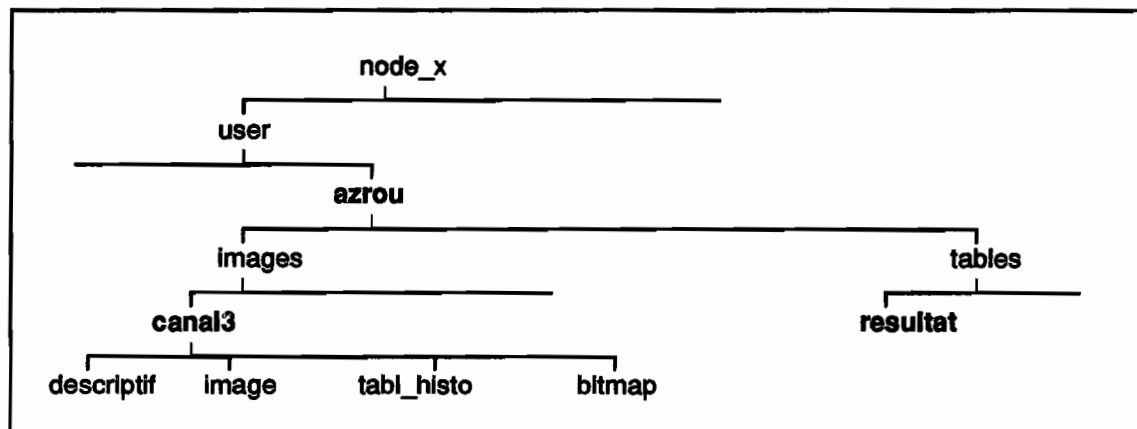


Figure 1.- Structure des fichiers images

I.B. Format image

C'est un format sophistiqué qui fait appel à une gestion du compilateur et du système d'exploitation.

Il est constitué de la manière suivante :

un en-tête

et une suite d'enregistrements, chacun d'entre-eux étant suivis de 4 octets indiquant quelle est la longueur de l'enregistrement.

Ceci permet au système de gérer le pointeur du fichier.

Ce type de format est facile à gérer en Fortran et en Pascal, mais relativement compliqué en langage C; il est en fait impossible de le gérer à partir des bibliothèques C, et doit se faire par appel système. En outre, il demande une place importante pour stocker l'image, puisqu'à chaque enregistrement sont associés 4 octets, plus l'en-tête de début. Il eut sans doute été plus astucieux de sauvegarder les images sous forme séquentielle puisque le descriptif indique la longueur des enregistrements.

Les canaux auraient été alors dans le format le plus simple qui puisse exister : pas d'en-tête, pas d'indicateur de fin de ligne. Ainsi, un canal de 1000 lignes de 1000 pixels aurait exactement 1000000 d'octets de longueur, ce qui signifie qu'à priori, on ne sait pas s'il y a 1000 lignes de 1000 pixels ou bien 500 lignes de 2000 pixels.

Les images sont consignées en format REC.

I.C. Descriptif

C'est le descriptif qui permet d'établir la différence. Ces descriptifs contiennent six enregis-

trements :

- (1) Nombre de lignes
- (2) Nombre de colonnes
- (3) Type de l'image
 - (a) image 8 bits : code 1
 - (b) image integer 4 : code 4
- (4) Dynamique : borne inférieure
- (5) Dynamique : borne supérieure
- (6) Taille du pixel carré (en décimètres)

A ces 6 premiers enregistrements devraient bientôt s'en ajouter deux autres relatifs à la position de la sous-scène sur l'image totale (première ligne et première colonne), dès qu'un nouveau programme de lecture de bande aura été mis au point.

Par ailleurs, le descriptif peut aussi contenir des indications supplémentaires relatives au déroulement de certains programmes (par exemple, PARPON ou PARSIFAL), ces indications permettant à l'utilisateur de vérifier la signification de l'image traitée.

I.D. Tableaux répertoriant les Contours

Contrairement à ce que nous avons fait dans le cadre du logiciel TAPIS (PARROT op. cit.), les contours détectés ne sont pas consignés sur une image, mais dans un tableau placé dans la branche //nom_noeud/user/NomRépertoire/tables, branche qu'il convient de créer dans le sous-répertoire //nom_noeud/user/NomRépertoire avant d'utiliser les programmes du logiciel ADONIS.

Le tableau des contours NomFich.TAB est créé par le programme ADONIS ou le programme ADONIS_VIS, le programme ASTICOT créant lui aussi un tableau NomFich.TAB qui associe n tableaux NomFich.TAB. C'est à partir de ces tableaux que les programmes aval de traitement tracent éventuellement les contours sélectionnés en créant une image.

Chaque contour peut en effet être représenté comme un ensemble d'éléments l_i

$$L = [l_i : i = 1, \dots, n]$$

où l_i correspond à l'ensemble des points du contour détecté.

Afin de limiter la place requise, les tableaux ont le format $(n(2a2))$, n correspondant au nombre d'enregistrements nécessaires pour décrire les contours. A cette différence près, ces tableaux ont la même structure que le tableau provisoire du programme SALAD (in TAPIS, PARROT op. cit.) : les points (x,y) d'un contour sont compris entre un doublet (0,z) indiquant le numéro d'ordre z du contour, et un doublet (0,0) signant la fin de chacun des contours.

De plus, le tableau comprend en en-tête un doublet fournissant la taille de l'image traitée (nombre_lignes, nombre_colonnes) qui permet de situer les coordonnées des points du contour. Des images peuvent ainsi être reconstituées si nécessaire.

La transformation de nombres en caractères obéit à la procédure suivante :

– soit ival1 et ival2, les valeurs du doublet à transformer,


```
integer*4 ival1,ival2,jval,kval
```

```
character*1 car(4)
```

```
jval=int(float(ival1)/100.)
```

```
car(1)=char(jval)
```

```
kval=ival1-(jval*100)
```

```
car(2)=char(kval)
```

```
jval=int(float(ival2)/100.)
```

```
car(3)=char(jval)
```

```
kval=ival2-(jval*100)
```

```
car(4)=char(kval)
```

```
write(unit,100) (car(i),i=1,4)
```

```
100 format(4a1)
```

Nota : char(10) correspondant à l'ordre "retour charriot", il convient de transformer la valeur 10 en 255, à l'écriture et à la lecture. Exemple de lecture :

```
ival=char(car(n))
```

```
if(ival.eq.255) ival=10
```

I.E. Tableaux répertoriant les variables discriminantes

Sur la branche //nom_noeud/user/NomRépertoire/tables se placent les fichiers de résultat consignnant les valeurs des variables discriminantes calculées pour chaque structure détectée par le programme ADONIS. Ce fichier porte le nom de l'image traitée, avec l'extension .RES.

Les tableaux NomFich.RES (cf. Tableaux II et III) sont formés par une succession d'enregistrements de format 99i5, indiquant, pour chaque structure détectée, son numéro d'ordre, ses coordonnées x et y, la valeur du rayon, le nombre de pixels formant le contour, le coefficient d'intersection, la direction de la normale, les coefficients de corde et de symétrie, la valeur angulaire de la pente de la normale à la corde, l'occurrence, les coordonnées x et y du début de la corde, les coordonnées x et y de la fin de la corde, et pour finir : soit la réédition du numéro d'ordre de la structure pour les tableaux directement issus du programme ADONIS, soit les numéros d'ordre des structures associées à l'occurrence pour des tableaux issus du programme CONCORDE, soit enfin les valeurs des n variables de surface pour les tableaux issus du programme CONCERT_VIS.

I.F. Tableaux de correspondance

Les programmes PRESTA et TRICOT assurant une sélection de structures parmi l'ensemble figurant dans les tableaux NomFich.RES et NomFich.TAB provenant du programme ASTICOT, un tableau de correspondance NomFich.COR est créé, afin d'aller rechercher directement les structures triées dans les tableaux d'origine.

I.G. Gestion des tableaux par programme

Le tableau ci-dessous illustre de manière synthétique les relations des différents programmes entre les images lues et/ou créées et les tableaux lus et/ou créés.

PROGRAMMES	IMAGES		TABLEAUX LUS			TABLEAUX CREES		
	LUES	CREES	.RES	.TAB	.COR	.RES	.TAB	.COR
ADONIS	+					+	+	
ASTICOT			+	+		+(1)	+(1)	
CAPNOR			+			+(2)		
CONCERT_VIS	+(3)		+	+		+		
CONCORDE			+			+(4)		
ECLAT_STAT		+	+	+	+			
HISBID_COR		+(5)	+		+			
HIST_COR		+(6)	+		+			
MIN_MAX			+		+			
ROSACE		+(9)	+		+			
TRICOT		+	+(7)	+				+
PRESTA			+					+(8)

(1) résultat de l'association de n tableaux .RES et .TAB

(2) CAPNOR lit NomFich.RES et réécrit sur ce fichier

(3) lecture de 1 à n fichiers images contenant les variables de "surface"

(4) CONCORDE lit un fichier NomFich1.RES et engendre un autre fichier NomFich.RES

(5) création d'un fichier image NomFich.HB

(6) création de fichiers images à extensions diverses : .RAY, .DIR, .PNC, etc...

(7) lecture de 2 fichiers NomFich.RES qui peuvent être identiques

(8) création d'un fichier .COR et de fichiers traités par STATITCF

(9) création d'un fichier image NomFich.ROS

Tableau I.- Gestion des tableaux par les différents programmes du logiciel.

II. TRAITEMENTS

La détection non supervisée des formes sub-circulaires sur les images satellitaires repose sur une mise en évidence de contours significatifs et sur la recherche des centres et rayons des éléments sub-circulaires contenus dans ces contours.

II.A- EXTRACTION DE CONTOURS SIGNIFICATIFS

(1) nature des données à traiter

Les structures présentes sur les images satellitaires à traiter proviennent :

- (a) soit de sauts de réflectance entre des objets divers;
- (b) soit de structures sécantes à différents objets.

Selon la nature de la texture de l'image, la mise en évidence et l'extraction de contours significatifs se font de manière différente.

Dans le premier cas (a), nous avons développé une algorithmique qui traite de manière objective les images. Seule, dans ce manuel, cette algorithmique sera décrite.

Dans le second cas (b), l'extraction des contours doit faire l'objet d'un traitement spécifique qui est en cours de développement.

(2) extraction des contours de type (a)

L'extraction des éléments structuraux que comporte une image satellitaire dépend étroitement de la texture rencontrée. Dans l'exemple servant à illustrer les résultats fournis par la méthode, les structures correspondent pour l'essentiel à des limites entre plages à réflectance plus ou moins homogènes, et non à la présence de traits structuraux traversant des plages à réflectance homogène.

En vue de lisser au mieux les formes rencontrées, l'image des données brutes est traitée par un filtre passe-bas de taille 3x3 (Programme FILTRES). Ce filtrage est réitéré jusqu'à l'obtention d'une forte stabilité. Pour ce faire, l'arrêt des itérations se produit quand le nombre de pixels modifiés au cours de la dernière itération est inférieur ou égal à 1 % du nombre de pixels modifiés au cours du tout premier filtrage (Fig. 3A). Il convient de noter que la forme de la courbe des pourcentages de modification, au cours des itérations successives, se révèle être un bon indicateur de la texture de l'image étudiée.

Le seuillage des valeurs de l'histogramme de l'image lissée se fait à partir d'un seuil de réflectance variable de valeur N, tel que le nombre de pixels dont les valeurs sont inférieures ou égales à ce seuil, corresponde à un pourcentage donné du nombre total de pixels que comprend l'image entière (Programme PARPON). On peut par exemple fixer 9 niveaux de pourcentage, allant de dix en dix, de 10% à 90%. Dans chacune des 9 scènes ainsi obtenues, les pixels correspondant sur l'image lissée aux pixels de réflectance inférieure ou égale au seuil N, sont codés avec la valeur 1, le fond étant mis à 0.

Chacune de ces images binaires est ensuite soumise à un lissage majoritaire (GOLDBERG *et al.* 1975, in SCHOWENGERDT 1983); le pixel central d'une fenêtre 3x3 prend la valeur présentant

l'occurrence maximum dans cette fenêtre. Ce lissage (Programme **FILTRES**) est réitéré jusqu'à stabilité complète de l'image. Appliqué à une image binaire, ce filtre se comporte comme un opérateur de Morphologie Mathématique (SERRA 1982, 1988, COSTER et CHERMANT 1985); il présente l'avantage d'ébarbuler les formes et d'en faire disparaître les pixels d'angle droit (Fig. 3C).

Enfin, les limites des différentes formes ainsi obtenues sont tracées en analysant les sauts de valeur entre deux pixels voisins, selon deux directions (NS et EW), afin d'obtenir un contour de connectivité 8 (SCHWEIZER 1987). En vue de limiter au maximum les croisements structuraux provenant parfois de la coalescence des formes, le saut de réflectance, c'est à dire la valeur absolue de la différence entre les codes de deux pixels voisins, est placé sur la valeur maximum rencontrée dans chaque couple de valeurs pendant le balayage de l'image, le contour de ces formes étant de la sorte situé sur le bord de la forme elle-même (Programme **LIMITE**; Fig. 3D).

L'image résultante est nettoyée à l'aide des programmes **DEST_ANG** et **DEST_CADRE**.

Cette séquence comprend une série de programmes qui s'enchaînent de la façon suivante :

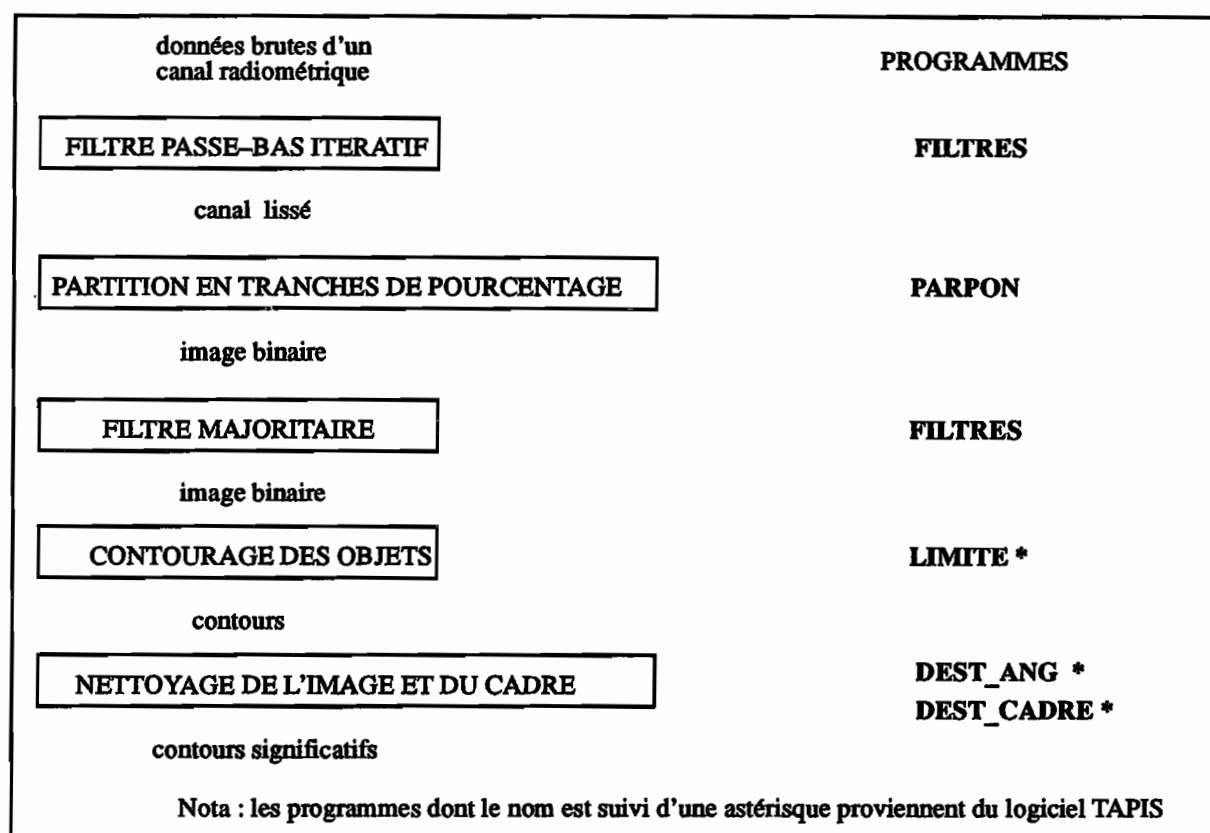


Figure 2.— Mise en évidence des contours de type (a).

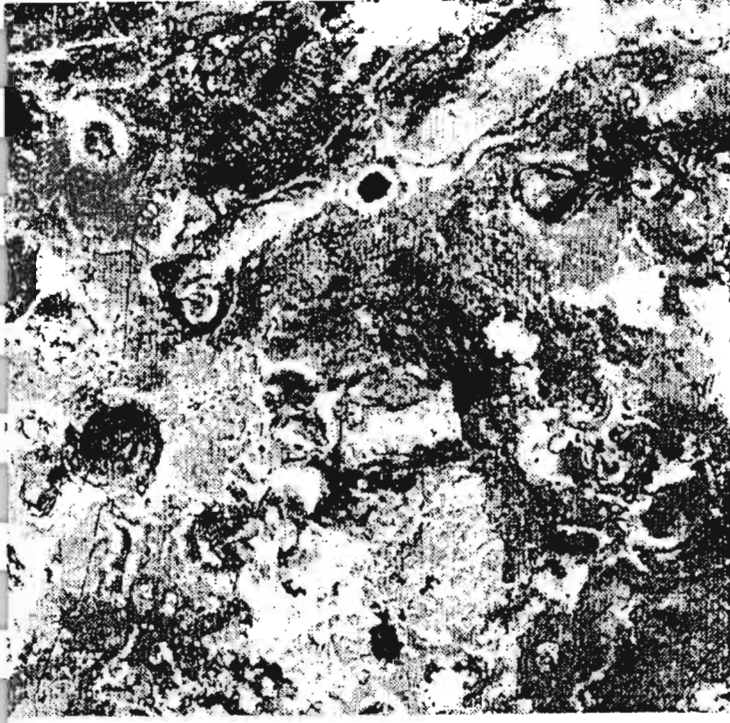
C'est sur les contours significatifs ainsi obtenus que s'effectue la détection des segments sub-circulaires.

En fait, le programme **PARSIFAL** rassemble en seul traitement les programmes **PARPON**, **FILTRES**, **LIMITE**, **DEST_ANG** et **DEST_CADRE**. De plus, le programme **PARSIFAL2** permet d'obtenir en une seule fois, toutes les images correspondant aux différents seuils de réflectance définis par les tranches de pourcentage.

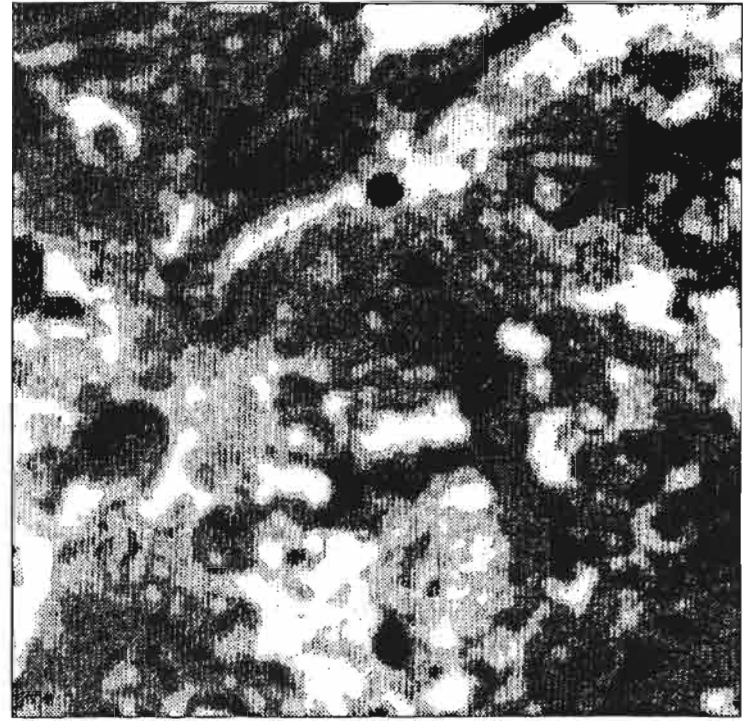
COSTER M. et CHERMANT J.-L. (1985).— Précis d'Analyse d'Images. CNRS, Paris, 521 p.

SCHWEIZER P. (1987).— Infographie I. Presses polytechniques romandes, Lausanne, 418 p.

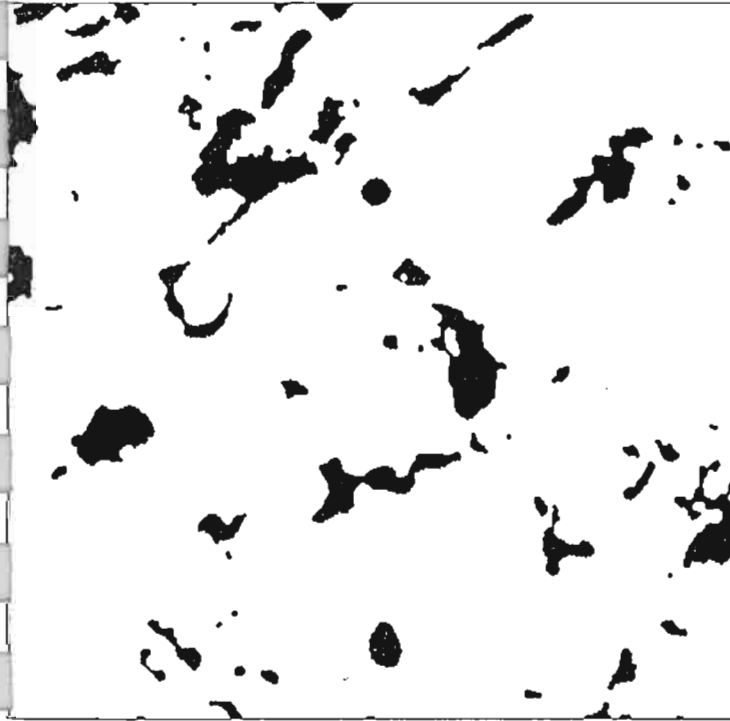
SERRA J. (1982, 1988).— Image Analysis and Mathematical Morphology. Academic Press Ed. London/New-York, 1033 p.



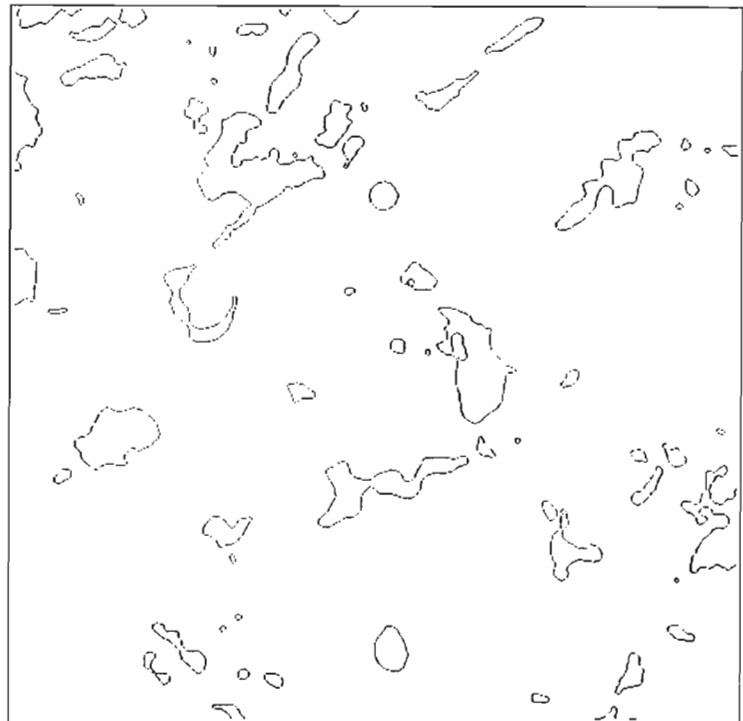
A



B



C



D

Figure 3.— Région test d'Azrou (Moyen Atlas marocain).
A : Canal XS3; B : Image lissée; C : Image binaire (10%); D : Contours.

(3) extraction des contours de type (b)

L'extraction des éléments structuraux de type (b) est notamment obtenue par l'utilisation des opérateurs de la Morphologie Mathématique (SERRA op. cit., COSTER et CHARMANT op. cit.). Quelques programmes de Morphologie Mathématique ont été partiellement développés dans le logiciel TAPIS (PARROT op. cit.), et nous avons aussi ajouté le programme ERODIL dans le logiciel ADONIS. De plus, il doit bien en traîner par ci par là quelques-uns dans les différents répertoires du réseau, mais ils sont toujours malheureusement toujours non documentés.

Toutefois, si ces traitements extraient effectivement les contours que comporte une image satellitaire, ils présentent souvent le désavantage de scinder en petits tronçons les contours en question.

En vue de remédier à cet inconvénient et de tenter de rassembler de manière objective et logique ces éléments de courbe, le programme COCON (CONNECTION de CONtours) a été mis au point.

Basé sur le principe du suivi de contours, il connaît, pour chaque élément structural rencontré, la position (x,y) des deux extrémités et la direction angulaire de ces extrémités. Il peut ainsi rechercher, pour un proximité donnée, les extrémités de deux contours voisins susceptibles d'être soudés.

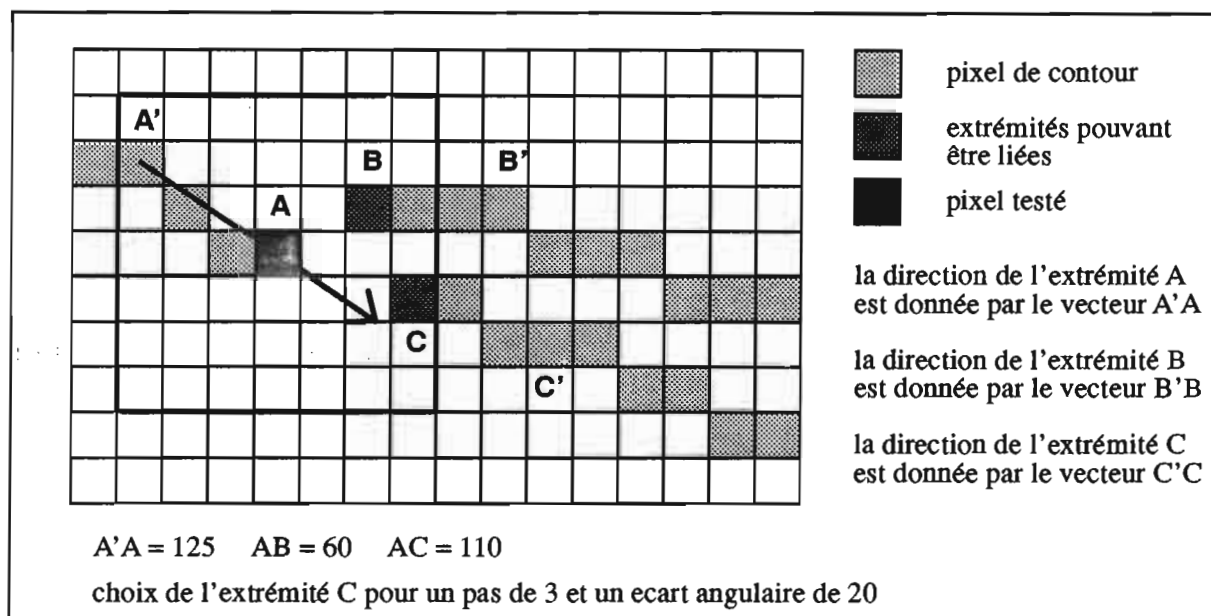


Figure 4.- Principe de fonctionnement du programme Cocon.

Par ailleurs, nous avons développé un programme (REGIME) qui squelettise une image binaire d'aplats en lignes de connectivité 8, remplissant d'emblée les conditions exigées par le programme ADONIS.

II.B.- DETECTION DES SEGMENTS SUB-CIRCULAIRES

Tous les traits structuraux (lignes droites ou courbes, sécantes ou isolées) provenant de l'ensemble des traitements d'extraction des contours, sont ensuite soumis aux traitements de détection et d'individualisation.

Le programme ADONIS (ou sa version ADONIS_VIS comportant la visualisation du traitement sur l'écran) représente le noyau principal du logiciel.

Ce programme comprend quatre sous-étapes qui s'enchaînent automatiquement les unes les

autres.

(1) individualisation des contours

Les courbes sont individualisées une à une par un suivi de contour basé sur le mouvement d'une fenêtre glissante (W_n) de 3x3 pixels, n indiquant la position de la fenêtre sur le n ième pixel du contour. L'algorithme développé permet de prendre une décision sur la direction à suivre, qu'un croisement se présente ou non. Les opérations de suivi de contour sont basées sur le code de Freeman (FREEMAN et DAVIS 1977).

(2) décomposition des courbes en éléments sub-circulaires.

Les contours individualisés à la sous-étape précédente sont parfois formés par une succession d'éléments sub-circulaires qui sont alternativement convexes et concaves.

Le contour est segmenté en éléments sub-circulaires quand il présente un changement de direction de courbure.

(3) application de la Transformée de Hough

Quand un élément sub-circulaire est individualisé à la sous-étape précédente, on applique ensuite une version de la Transformée de Hough (ILLINGWORTH et KITTLER 1987) afin de déterminer la position approximative du centre de cet élément et la valeur approchée de son rayon.

La Transformée de Hough (HOUGH 1962) convertit un problème complexe de détection de formes dans l'espace image en une détection de pics plus facile à résoudre dans un espace paramétrique. Cet espace est quantifié et utilisé comme accumulateur.

(4) calcul de la position exacte du centre

Avec la méthode de Hough, la détection des structures circulaires repose essentiellement sur les pixels formant le contour. Une autre approche consiste à considérer tous les pixels qui n'appartiennent pas au contour comme autant de centres probables de ce contour.

Le cercle ainsi détecté que nous appellerons Cercle Référentiel (CR) et qui satisfait les conditions expérimentales retenues (par exemple, contour de 50 pixels minimum, segment de 30 pixels minimum, intersection de plus de 20%, etc...), correspond au cercle qui englobe le maximum de pixels du contour étudié. Il peut contenir l'ensemble des pixels du contour, inscrire le contour ou bien être inscrit dans ce contour.

II.C.- RESULTATS

Quand la position exacte du centre (x,y) et la valeur précise du rayon (R) du Cercle Référentiel (CR) sont connues, l'algorithme mis au point fournit pour chaque élément sub-circulaire détecté, outre les coordonnées du centre et la valeur du rayon de ce cercle, des informations complémentaires permettant de mieux individualiser la structure détectée.

FREEMAN H. and DAVIS L.S. (1977).— A corner-finding algorithm for chain coded curves. *IEEE Trans. Comput.*, vol C-26 : 297-303.

HOUGH P.V.C. (1962).— A method and means for recognizing complex patterns. U.S. Patent 3,069,654.

ILLINGWORTH J. and KITTLER J. (1987).— The adaptive Hough Transform. *IEEE Trans. PAMI*, 9, n 5 : 690-698.

Toutes ces variables sont reportées dans un tableau (cf. Tableau II, page 18) qui comprend

respectivement : le numéro d'ordre de la structure détectée, la position x et y du centre du Cercle Référentiel (CR), la valeur du rayon (R) de ce cercle, l'ensemble des variables npcd, CI, DIR, CC, DNC, l'occurrence OCC, les positions (x,y) du début de la corde, les positions (x,y) de la fin de la corde et le ou les numéros d'ordre des structures correspondant à l'occurrence (OCC).

On obtient un tableau pour chacune des images correspondant au seuillage en pourcentage décrit ci-dessus. Ces tableaux sont rassemblés en un seul tableau de même type (Programme ASTICOT), qui contient ainsi toutes les données relatives à toutes les structures détectées.

(1) définition des variables de "contour"

(a) npcd : nombre de pixels du contour détecté.

(b) CI : coefficient d'intersection. Il s'agit du pourcentage de pixels de CR s'intersectant (npi) avec les pixels du contour étudié (npcd); $CI = (npi/npcd) * 100$. Dans l'algorithme développé, le contour n'est pris en compte que pour un CI supérieur ou égal à 20. Plus ce coefficient est élevé, plus le contour décrit un cercle ou une portion de cercle parfait.

(c) DIR : direction de la normale à la corde de l'élément détecté. Cette normale à la corde passe par le centre (x,y) de CR. Les valeurs sont en degrés et sont codées dans le sens de la marche des aiguilles d'une montre, l'origine (0) étant situé au nord de l'image; il convient de noter que, pour une image satellitaire, la direction géographique réelle correspond à cette valeur plus l'angle de trajectoire du satellite. Par convention, nous avons fixé à 1000 cette valeur pour une structure fermée pour laquelle la notion de corde et de normale à la corde est sans signification.

(d) CC : coefficient de corde. $CC = (ln/2R) * 100$, ln étant égal à la distance entre l'intersection de la normale avec la corde et l'intersection de la normale avec le contour. Ce coefficient permet d'appréhender la forme du contour étudié. Un coefficient inférieur à 50 rend compte d'un arc inférieur au demi-cercle, un coefficient supérieur à 100 signifiant que le cercle CR est inscrit dans le contour détecté. Par convention, le coefficient d'une structure fermée a été fixé à 100, valeur correspondant à un cercle complet, et cela, quel que soit le CI de cette forme.

(e) DNC : c'est le rapport multiplié par 100 de la distance entre l'intersection de la normale avec la corde et le milieu de cette corde, divisé par la valeur du rayon (R). Ainsi, une valeur de DNC égale ou proche de 0, indiquera-t-elle que la structure sub-circulaire est symétrique.

(f) OCC : il s'agit de l'occurrence qui sur le tableau I est, par définition, égale à 1.

(2) notion d'occurrence

L'image traitée ayant été scindée en n images en fonction du pourcentage, une même structure sub-circulaire peut être mise en évidence à plusieurs reprises. Ceci est dû aux phénomènes suivants :

(a) une forme mise en évidence pour un seuil de réflectance dépendant du pourcentage de segmentation retenu, peut se maintenir, avec ou sans dilatation, pour le ou les seuils de réflectance d'autres tranches de pourcentage, sans être ennoyée dans une nouvelle forme.

(b) l'un ou l'autre des niveaux de seuillage peut faire ressortir une section différente appartenant à une même structure.

De plus, la décomposition en éléments sub-circulaires effectuée sur les limites des formes peut scinder en plusieurs arcs de cercle une même structure circulaire.

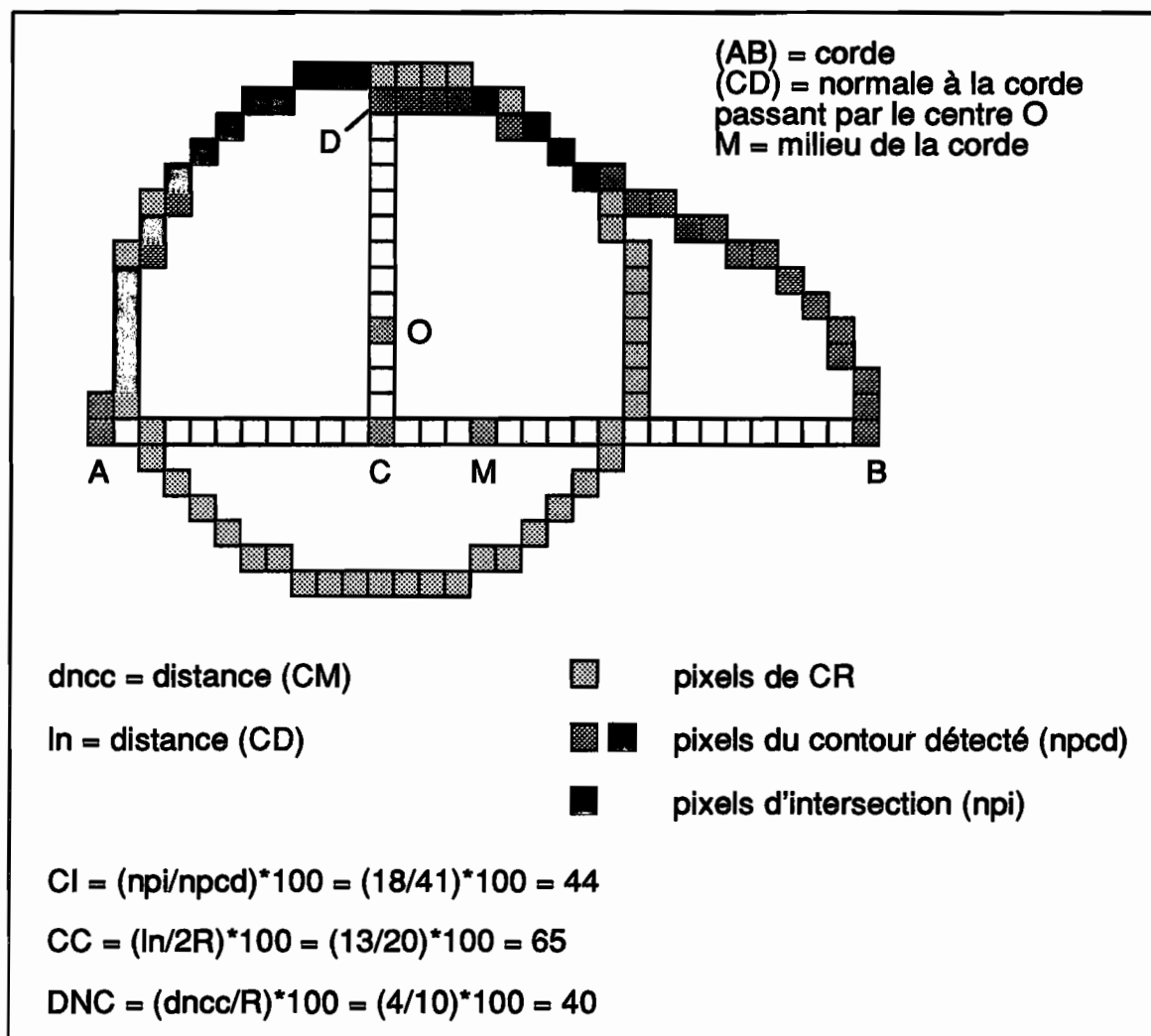


Figure 5.- Les variables de contour

Ainsi, lorsque l'on assemble les n tableaux, il est fréquent d'obtenir, pour des éléments structuraux différents, des centres et des rayons sensiblement identiques. A ce stade, on recherche, pour un intervalle donné valable pour x , y et R , le nombre d'éléments structuraux dont le centre et le rayon sont compris dans cet intervalle. Ce nombre correspond à l'occurrence (OCC). A cette occurrence, reportée sur le Tableau résultant (cf. Tableau III, page 19), sont associés le ou les numéro d'ordre des structures concernées figurant sur le premier Tableau (cf. Tableau II, page 18).

Ceci permet, lors des traitements ultérieurs effectués à partir du Tableau II, d'aller rechercher dans le premier Tableau, les caractéristiques propres de chacune des structures considérées comme formant un seul élément sur le second Tableau.

Par ailleurs, la recherche de l'occurrence pour un intervalle donné, entraîne un certain nombre de modification de variables, à savoir : x , y , R et DIR . Par convention, et pour conserver aux deux tableaux la même disposition (ce qui permet éventuellement de faire directement des sélections à partir du premier Tableau, les autres variables, non utilisées dans le second, sont portées à zéro.

$$x = \sum x_i / occ \quad (i = 1, \dots, occ)$$

$$y = \sum y_i / occ \quad (i = 1, \dots, occ)$$

$$r = \sum r_i / occ \quad (i = 1, \dots, occ)$$

$$\begin{aligned}
\text{dir}_{\text{occ}} &= (\text{dir}_i + \text{dir}_{i+1})/2 & \text{si } |\text{dir}_{i+1} - \text{dir}_i| < 180^\circ \\
&= (\text{dir}_i + \text{dir}_{i+1})/2 - 180^\circ & \text{si } |\text{dir}_{i+1} - \text{dir}_i| > 180^\circ \\
&= 2000 \text{ par convention,} & \text{si } |\text{dir}_{i+1} - \text{dir}_i| = 180^\circ \\
& & i = 1, \dots, \text{occ}-1
\end{aligned}$$

	x	y	r	nbpix	CI	DIR	CC	DNC	PNC	OCC					
1	41	215	22	70	32	303	31	73	0	1	77	181	27	214	1
2	69	55	25	54	22	338	15	44	0	1	58	40	42	80	2
3	123	158	26	58	31	277	19	35	0	1	140	141	84	148	3
4	145	155	11	51	38	69	95	18	0	1	140	142	161	150	4
5	221	136	20	98	34	196	95	45	0	1	209	156	195	108	5
6	207	132	25	55	52	177	44	24	0	1	212	121	210	156	6
7	371	263	22	79	26	344	22	45	0	1	370	216	351	282	7
8	366	146	22	51	40	155	28	5	0	1	372	159	380	142	8
9	461	269	14	80	36	1000	100	0	0	1	465	256	466	256	9
10	45	336	8	62	25	320	42	238	0	1	49	329	6	380	10
11	136	260	11	61	71	1000	100	0	0	1	125	260	125	259	11
12	209	151	28	50	41	275	23	4	0	1	230	133	183	137	12
13	204	148	38	56	35	220	13	3	0	1	240	148	209	111	13
14	221	163	17	51	42	140	23	29	0	1	230	168	220	180	14
15	201	285	17	63	31	23	137	35	0	1	198	270	215	311	15
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
118	395	180	9	52	42	1000	100	0	0	1	399	175	399	175	118
119	491	304	16	76	32	1000	100	0	0	1	480	309	480	308	119

Tableau II.- Représentation partielle des données de toutes les caractéristiques des 119 structures détectées sur l'image test d'Azrou.

Par convention, si l'une des directions rencontrée est égale à 2000 (c'est à dire, structure fermée), la direction résultante (dir_{occ}) est portée à cette même valeur. Par ailleurs, si deux directions sont opposées, la valeur de dir_{occ} consignée sur le Tableau résultant est de 3000, la structure étant considérée comme étant un cercle formé de traits discontinus.

(3) rapport des variables aux formes

Les variables qui caractérisent les structures détectées et/ou leur combinaison, permettent de discriminer les formes rencontrées.

Ainsi, un contour fermé dont le CC est par définition égal à 100, est régulier quand son CI est fort, et irrégulier quand son CI est faible. Une portion elliptique pourra avoir un CI et un CC faibles ou un CI moyen et un CC supérieur à 100. Un demi arc de cercle présente un CC voisin de 50, un CC supérieur à 100 indiquant que CR est inscrit dans le contour détecté. Toutes ces formes seront d'autant plus symétriques que DNC sera faible. Par ailleurs, les structures sub-rectilignes se caractérisent par un CI et un CC faibles.

	x	y	r	nbpix	CI	DIR	CC	DNC	OCC	(x,y) debut	(x,y) fin	Numéros d'ordre						
1	460	271	18	-	-	1000	-	-	2	-	-	-	9	77				
2	136	260	12	-	-	1000	-	-	5	-	-	-	11	29	39	53	67	
3	209	151	32	-	-	210	-	-	3	-	-	-	12	13	25			
4	310	86	24	-	-	75	-	-	4	-	-	-	19	31	44	73		
5	351	207	18	-	-	85	-	-	5	-	-	-	20	42	51	69	85	
6	467	266	30	-	-	189	-	-	3	-	-	-	23	34	46			
7	39	172	15	-	-	211	-	-	2	-	-	-	26	36				
8	41	4	32	-	-	90	-	-	2	-	-	-	27	37				
9	174	481	21	-	-	190	-	-	2	-	-	-	30	40				
10	304	309	14	-	-	63	-	-	4	-	-	-	41	56	87	101		
11	323	386	25	-	-	252	-	-	6	-	-	-	43	55	72	88	102	114
12	171	202	27	-	-	335	-	-	2	-	-	-	48	99				
13	389	157	30	-	-	290	-	-	2	-	-	-	52	70				
14	315	78	35	-	-	82	-	-	2	-	-	-	58	84				
15	-6	317	33	-	-	153	-	-	2	-	-	-	61	78				
16	58	464	31	-	-	170	-	-	3	-	-	-	64	79	95			
17	98	63	26	-	-	96	-	-	3	-	-	-	65	66	96			
18	416	390	18	-	-	177	-	-	2	-	-	-	76	92				
19	189	57	32	-	-	303	-	-	2	-	-	-	83	100				
20	358	53	15	-	-	329	-	-	3	-	-	-	86	105	116			
21	490	304	19	-	-	1000	-	-	2	-	-	-	110	119				

Tableau III.- Données des caractéristiques des structures à occurrence supérieures ou égales à 2, pour un intervalle de 10, et numéros d'ordre des structures "primaires" associées.

(4) les variables de "surface"

De plus, les valeurs de toute nature (données SPOT, données issues de combinaisons de canaux, codes morphologiques provenant de calculs effectués sur le Modèle Numérique de Terrain de la zone étudiée, etc...) rencontrées dans la surface qu'engendre chacun de ces cercles ou portions de cercle, peuvent être considérées comme des variables pouvant être prises en compte par une analyse statistique. Ces variables supplémentaires permettent ainsi de discriminer les formes, non seulement à partir de leurs contours, mais aussi à partir de leurs surfaces.

II.D.- ANALYSES DES VARIABLES

Une série de programmes permet d'exploiter ces résultats.

Tout d'abord, CAPNOR (CALcul de la Pente de la NORMale) calcule la pente de la normale à la corde, si l'on dispose du Modèle Numérique de Terrain correspondant à l'image traitée. La variable obtenue peut être traitée au même titre que les variables de contour ou de surface.

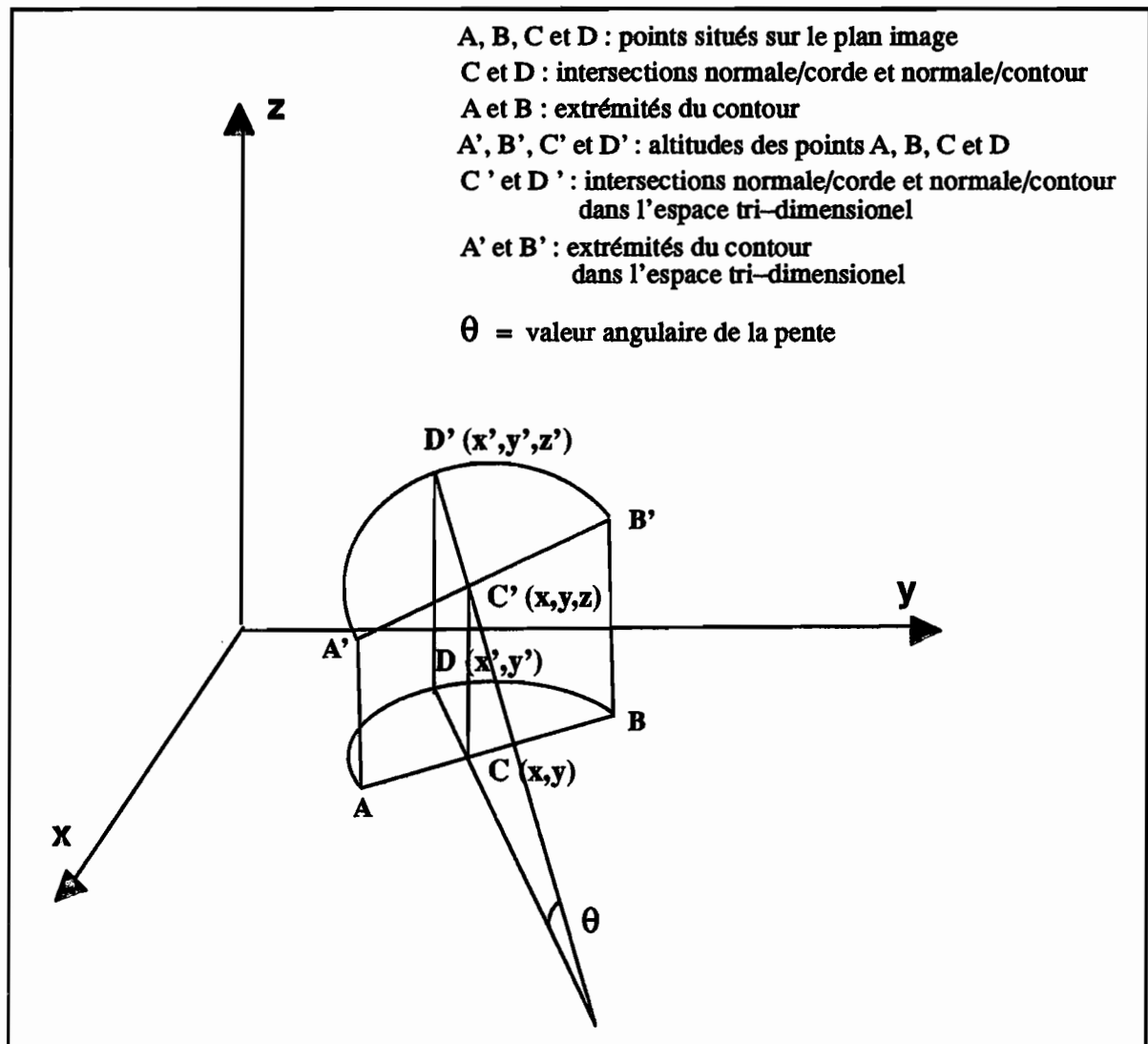


Figure 6.— Calcul de le Pente de la Normale à la Corde (PNC).

ASTICOT (ASsemblage des Tableaux Indexant les COURbes Traitées) assemble les n tableaux obtenus à partir des n images définies par les programmes PARSIFAL ou PARSIFAL2 (voire la suite PARPON, FILTRES, LIMITE, DEST_ANG et DEST_CADRE). C'est sur les tableaux issus de ce programme (NomFich.RES et NomFich.TAB placés dans //nom_noeud/user/NomRépertoire/tables) que s'appuient les autres programmes du module "Analyse".

CONCERT_VIS (CONtenu des CERcles Traités) assure l'extraction des valeurs moyennes ou dominantes rencontrées sur différents canaux dans les surfaces qu'engendrent les structures détectées.

CONCORDE (calcul de l'occurrence par contrôle des coordonnées).

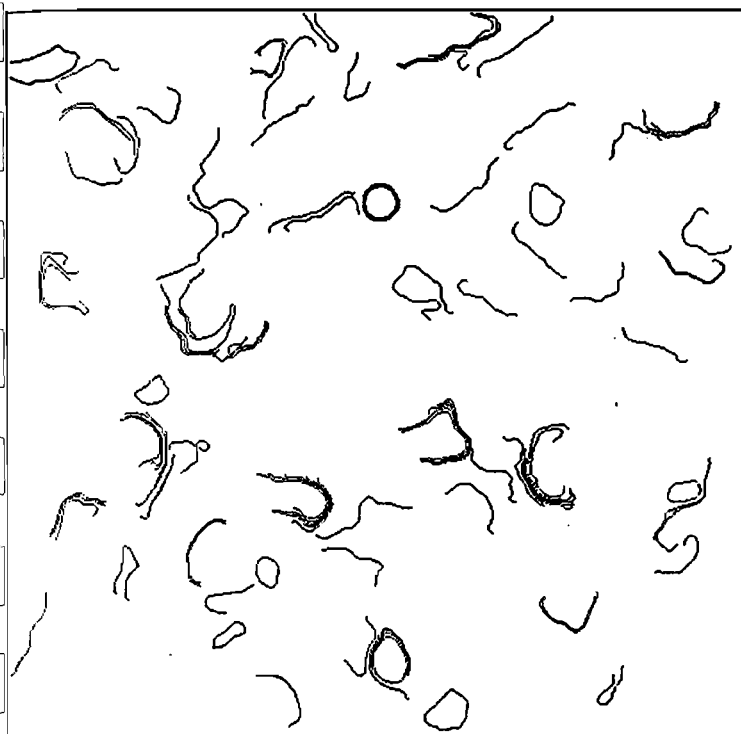
ECLAT_STAT (extraction des classes issues du traitement statistique)

HISBID_COR (Histogramme bi-dimensionnel DIR/PNC).

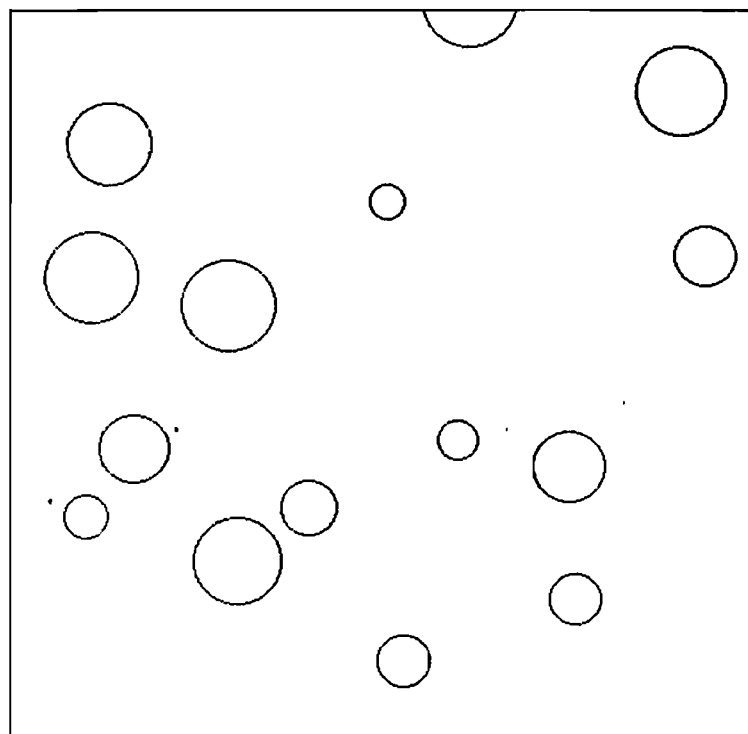
HIST_COR (Histogrammes sur toutes les variables).

ROSACE (rosaces des directions)

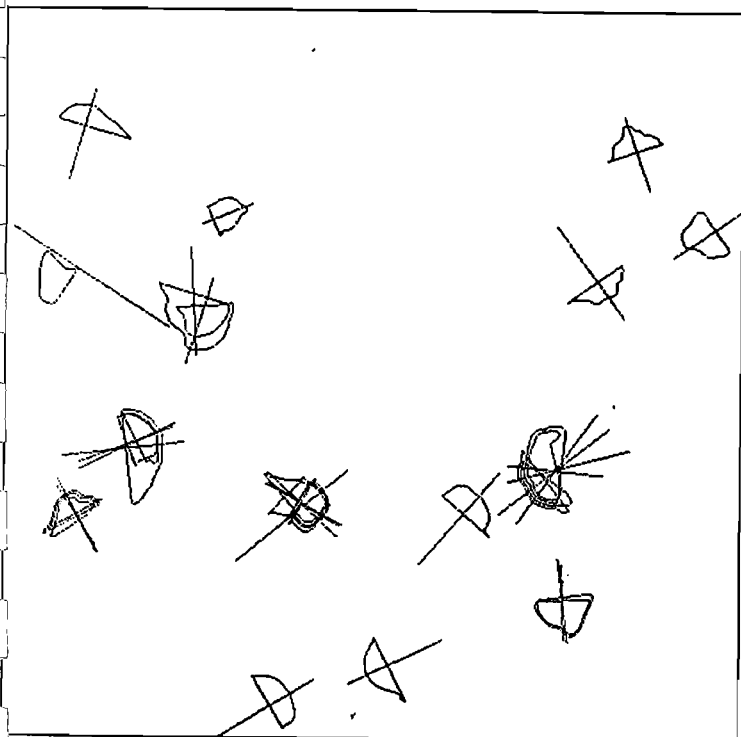
TRICOT (TRI des COURbes Traitées) permet de trier les structures détectées en fonction, soit des variables discriminantes de forme ou de surface, soit en appelant directement les numéros d'ordre après analyse statistique.



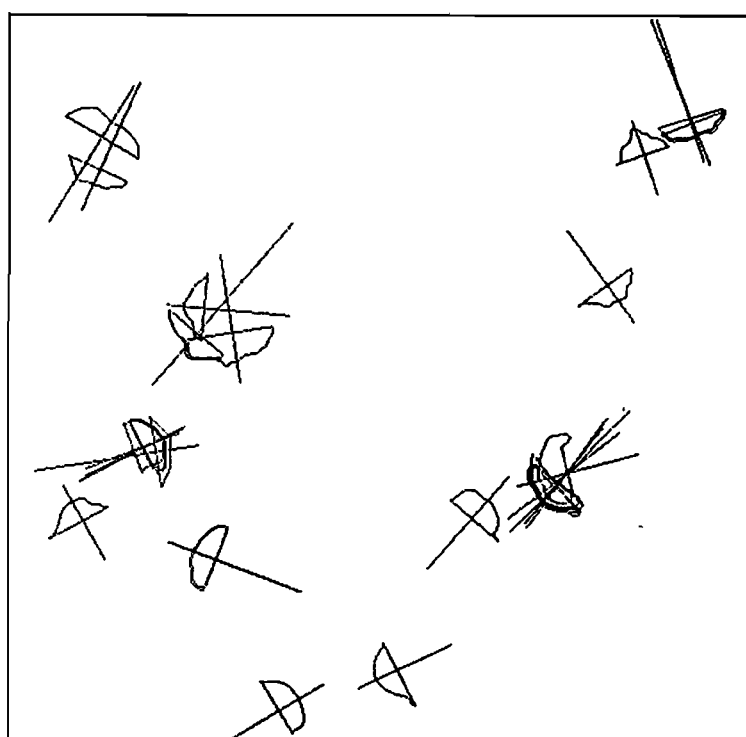
A



B



C



D

Figure 7.- Illustration de quelques résultats.

A : Structures détectées; B : Cercles pour une occurrence égale ou supérieure à 2;

C : Choix de l'utilisateur; D : Tri obtenu par analyse statistique.

II.E.- STATISTIQUES

Deux programmes assurent l'utilisation du logiciel STATICF sur micro-ordinateur. Les transferts nécessitent l'utilisation d'un programme Apollo (PRESTA) et d'un programme sur le micro-ordinateur (PREPSTAT).

II.F.- RESTITUTIONS GRAPHIQUES

PRELAS_A code les structures en une palette binaire utilisable par l'imprimante laser.

III.- FICHES PROGRAMME

Les Fiches Programme détaillées dans ce sous-chapitre rappellent en quelques lignes quelles sont les fonctions de chacun des programmes, le type de l'appel, et le Menu, c'est à dire la succession des questions posées par le programme, ainsi que des notas destinés à apporter des précisions sur le déroulement du programme.

Ces Fiches Programmes sont rangées par ordre alphabétique.

PROGRAMME ADONIS
PROGRAMME ADONIS_VIS

FONCTION :

Le programme ADONIS assure la détection des éléments sub-circulaires à partir des images binaires ne contenant que des contours.

Les résultats sont consignés dans deux tableaux NomFich.RES et NomFich.TAB que le programme crée dans le répertoire //nom_noeud /user/NomFich/tables. NomFich.RES contient les valeurs des variables de contour (cf. Table II, page 18); sur NomFich.TAB sont reportées les coordonnées de chacun des pixels constituant les contours extraits.

APPEL : ADONIS

MENU :

Q1 COMPTEUR ?

Q2 CODE ALPHA ?

Q3 CODE NUMERIQUE ?

Nota : Q1, Q2 et Q3 correspondent aux mots de passe nécessaires pour utiliser le programme ADONIS.

Q4 CHOIX DU NOEUD :

Q5 NOM DU FICHIER IMAGE A TRAITER ?

Q6 DANS LA DIRECTORY ?

Q7 SEUIL DES PIXELS SIGNIFICATIFS POUR UN CONTOUR (seuil ≥ 50) ?

Q8 SEUIL DES PIXELS SIGNIFICATIFS POUR UN ELEMENT CIRCULAIRE (seuil ≥ 20)?

Nota : Le programme ADONIS_VIS présente le même Menu que ADONIS. Il permet de plus de suivre en mode graphique le déroulement du programme.

PROGRAMME ASCOR

FONCTION :

Le programme **ASCOR** associe les tableaux **NomFich.COR** issus du programme **TRICOT** (cf. plus loin), en vue de regrouper des tranches de résultats différents et de les traiter comme formant un seul ensemble avec les programmes utilisant ce type de fichier.

APPEL : ASCOR

MENU :

- Q1** CHOIX DU NOEUD :
- Q2** NOM DE LA DIRECTORY OU SONT OBLIGATOIREMENT
SITUES LES FICHIERS A TRAITER ?
- Q3** NOM GENERIQUE DU PREMIER TABLEAU .COR ?
- Q4** NOM GENERIQUE DU DEUXIEME TABLEAU .COR ?
- Q5** NOM GENERIQUE DU TABLEAU DE SORTIE .COR ?

PROGRAMME ASTICOT

FONCTION :

Le programme ASTICOT (ASsociation de Tableaux Indexant les COurbes Traitées) associe les n tableaux résultats NomFich.RES et NomFich.TAB issus du programme ADONIS, en créant deux nouveaux tableaux NomFich.RES et NomFich.TAB

APPEL : ASTICOT

MENU :

Q1 CHOIX DU NOEUD :

Q2 NOM DE LA DIRECTORY OU SONT OBLIGATOIREMENT
SITUES TOUS LES TABLEAUX (extension .RES et .TAB) ?

Q3 NOMBRE DE FICHIERS A TRAITER ?

Q4 NOM GENERIQUE DU FICHIER n ?
Nota : le programme gère lui-même les extensions .RES et .TAB

La questions Q4 est répétée n fois en fonction de la réponse faite en Q3.

Q5 NOM GENERIQUE DU FICHIER A SAUVEGARDER ?

PROGRAMME CAPNOR

FONCTION :

Le programme **CAPNOR** (**CAL**cul de la **Pente** de la **NOR**male à la corde) calcule la valeur angulaire de la normale à la corde, si l'on dispose du Modèle Numérique de Terrain correspondant. CAPNOR prend le relais du programme ADONIS, ajoutant au tableau .RES les valeurs de l'angle de la pente.

APPEL : CAPNOR

MENU :

- Q1 **CHOIX DU NOEUD :**
- Q2 **NOM DE LA DIRECTORY OU SONT OBLIGATOIREMENT
SITUES TOUS LES FICHIERS A TRAITER (tables et images à lire et à créer) ?**
- Q3 **NOM GENERIQUE DES TABLEAUX NomFich.RES ET MomFich.TAB
CORRESPONDANTS AUX RESULTATS DU PROGRAMME ADONIS ?**
- Q4 **NOM DU MNT (en integer 4) A TRAITER ?**

PROGRAMME COCON
PROGRAMME COCON_VIS

FONCTION :

Le programme COCON (COnnection des CONtours par leurs extrémités) permet de coller des contours en tenant compte du sens des directions de leurs extrémités et de la distance entre ces extrémités.

APPEL : COCON

MENU :

Q1 CHOIX DU NOEUD :

Q2 FICHIER ENTREE

NOM DU FICHIER ?

Q3 NOM DU REPERTOIRE ?

LE FICHIER SORTIE S'APPELERA :
NomFich_entree.COL

Q4 PRISE EN COMPTE DE LA DISTANCE MAXIMALE
ENTRE DEUX DE CONTOUR A COLLER :
0 ———> OUI 1 ———> NON

si OUI en Q4, la question Q5 est posée;
si NON, la distance maximale est fixée à 100; passage à Q6.

Q5 VALEUR DE LA DISTANCE MAXIMUM ?

Q6 CHOIX DE L'ANGLE ENTRE DEUX CONTOURS A COLLER :
0 ———> 90 1 ———> 180

Nota : Le choix 1 signifie que l'on cherche à coller des extrémités de sens opposé. Le choix 0 cherche à associer des extrémités perpendiculaires.

Q7 PRISE EN COMPTE DE L'ECART ANGULAIRE :
0 ———> OUI 1 ———> NON

si OUI en Q7, la question Q8 est posée;
si NON, l'écart angulaire est fixé à 10; passage à Q9.

Nota : On entend par écart angulaire, la différence maximum entre l'angle que fait la direction de l'extré-

mité du contour traité et celle des extrémités des contours voisins. Seules les extrémités candidates comprises dans cet écart sont soumises aux tests de sélection.

Q8 **VALEUR DE L'ECART ANGULAIRE ?**

Q9 **PRISE EN COMPTE DU PAS POUR LA RECHERCHE
DU SENS DE LA DIRECTION DES EXTREMITES :**
0 ———> OUI 1 ———> NON

si OUI en Q9, la question Q10 est posée;
si NON, le pas est fixé à 5.

Q10 **VALEUR DU PAS ?**

Nota : Le pas correspond au nombre de pixels pris en compte pour calculer la direction de toutes les extrémités.

NOTA : Le programme COCON_VIS présente le même Menu que COCON. Il permet de plus de suivre en mode graphique le déroulement du programme.

PROGRAMME CONCERT_VIS

FONCTION :

Le programme **CONCERT_VIS** (CONtenu des CERcles Traités) est l'un des programmes aval de la recherche des structures circulaires. Il permet d'extraire à partir de n canaux de nature différente, les variables de "surface" associées aux structures détectées par ADONIS. Ce programme engendre un fichier .RES semblable à celui créé par ADONIS, mais où figurent à la fin de chaque enregistrement, les valeurs des n variables de "surface".

APPEL : CONCERT_VIS

MENU :

- Q1 CHOIX DU NOEUD :
- Q2 NOM DE LA DIRECTORY OU SONT OBLIGATOIREMENT
SITUES LES FICHIERS A TRAITER ?
- Q3 NOM GENERIQUE DES TABLEAUX (NomFich.TAB et .RES) ?
- Q4 NOM GENERIQUE DU FICHIER A SAUVEGARDER ?
Nota : le programme gère lui-même les extensions .RES et .TAB
- Q5 NOMBRE DE FICHIERS IMAGE TRAITES ?
- Q6 NOM DU FICHIER n ?
- Q7 TYPE D'EXTRACTION :
 - 0 ———> VALEUR DOMINANTE
 - 1 ———> VALEUR MOYENNE

Les questions Q6 et Q7 sont répétées n fois en fonction de la réponse faite en Q5.

Nota : Le programme s'exécute en vérifiant que tous les fichiers traités ont bien la même taille.

PROGRAMME CONCORDE

FONCTION :

Le programme **CONCORDE** (**CON**trôle des **COo**RDonné**Es**) est l'un des programmes aval de la recherche des structures circulaires. Il permet de vérifier les correspondances entre les centres des arcs de cercle détectés par le programme **ADONIS** et de les assembler pour des seuils définis par l'utilisateur. Il intervient donc en aval du programme **ASTICOT**.

APPEL : CONCORDE

MENU :

- Q1** **CHOIX DU NOEUD :**
- Q2** **NOM DE LA DIRECTORY OU SONT OBLIGATOIREMENT
SITUES LES FICHIERS A TRAITER (tables) ?**
- Q3** **NOM GENERIQUE DU TABLEAU DE COORDONNEES DES CERCLES ?**
- Q4** **NOM GENERIQUE DU TABLEAU RESULTAT ?**
- Q5** **INTERVALLE DE CONFIANCE ?**

Nota : On entend par intervalle de confiance, l'intervalle que l'utilisateur définit pour que deux cercles soient considérés comme équivalents. Cet intervalle concerne les lignes et les colonnes et la valeur du rayon.

- Q7** **OCCURENCE (superieure ou egale) ?**

PROGRAMME DEST_ANG
PROGRAMME DEST_CADRE

FONCTION :

Ces deux programmes issus du logiciel TAPIS permettent de nettoyer sélectivement une image binarisée dans la dynamique (0,1) et comportant des traits structuraux.

APPEL : DEST_ANG
DEST_CADRE

MENU :

- Q1 CHOIX DU NOEUD :
- Q2 NOM DE L'IMAGE A TRAITER ?
- Q3 DANS LE DIRECTORY ?
- Q4 NOM DE L'IMAGE A CREER ?
- Q5 DANS LE DIRECTORY ?

PROGRAMME ECLAT_STAT

FONCTION :

Le programme ECLAT_STAT (Extraction des CLasses Après Traitement STATistique) extrait et visualise les structures appartenant aux différentes classes obtenues par traitement statistique.

APPEL : ECLAT_STAT

MENU :

- Q1 CHOIX DU NOEUD :
- Q2 NOM DE LA DIRECTORY OU SONT OBLIGATOIREMENT
SITUES LES FICHIERS A TRAITER ?
- Q3 NOM GENERIQUE DES TABLEAUX NomFich.TAB, NomFich.RES et
NomFich.COR ASSOCIES A NomFich ?
Nota : le nom générique du tableau .COR doit être identique à celui de .TAB et .RES
- Q4 NOM GENERIQUE DE L'IMAGE A CREER ?
- Q5 TRAITEMENT SUR LE TABLEAU DE CORRESPONDANCE :
0 ———> OUI 1 ———> NON
- Q6 NOMBRE DE CLASSES ?
- Q7 NOMBRE D'ELEMENTS DANS LA CLASSE n ?
- Q8 NUMERO DE L'ELEMENT m DANS LA CLASSE n ?

Nota : Q7 se répète en fonction de la réponse en Q6, et Q8 en fonction de la réponse en Q7.

- Q9 IL EST DE PLUS POSSIBLE DE :
(A) TRACER LES CONTOURS DETECTES AVEC OU SANS CORDE ET NORMALE
(B) TRACER LES CERCLES CORRESPONDANTS AUX STRUCTURES DETECTEES

INDIQUEZ VOTRE CHOIX :

- 1 ———> TRACER LES CONTOURS DETECTES
2 ———> TRACER LES CERCLES

Nota : si 1 en Q9, question Q10; si 2 en Q9, question Q11.

- Q10 1 ———> CONTOURS DETECTES
2 ———> CONTOURS (+ CORDE ET NORMALE)
- Q11 [P]érimètre [S]urface

PROGRAMME EQUIPOP

FONCTION :

Le programme EQUIPOP découpe l'image en n thèmes de population équivalente.

APPEL : EQUIPOP

MENU :

- Q1 CHOIX DU NOEUD :
- Q2 NOM DU FICHIER ?
- Q3 NOM DIRECTORY ?
- Q4 NOM DE L'IMAGE A CREER ?
- Q5 DANS LE DIRECTORY ?
- Q6 NOMBRE DE LOTS ?

PROGRAMME ERODIL

FONCTION :

Le programme **ERODIL** (**ERO**sion **DIL**atation) est un module de Morphologie Mathématique en multi-niveaux, utilisant un élément structurant carré de taille variable.

APPEL : ERODIL

MENU :

- Q1 CHOIX DU NOEUD :
- Q2 NOM DE L'IMAGE A TRAITER ?
- Q3 DANS LE DIRECTORY ?
- Q4 TAILLE DE LA MAILLE ?
- Q5 1 ———> DILATATION
 2 ———> EROSION
 3 ———> FERMETURE (dilatation + érosion)
 4 ———> OUVERTURE (érosion + dilatation)
 5 ———> FILTRE + (ouverture + fermeture)
 6 ———> FILTRE – (fermeture + ouverture)

après traitement :

- Q6 1 ———> RETOUR AU MENU (QUESTION 5)
 2 ———> SAUVEGARDE

PROGRAMME FILTRES

FONCTION :

Le programme **FILTRES** est un utilitaire qui permet d'effectuer divers types de filtrage.

APPEL : FILTRES

MENU :

Q1 CHOIX DU NOEUD :

Q2 NOM DE L'IMAGE A TRAITER ?

Q3 DANS LE DIRECTORY ?

Q4 TYPE DE FILTRE :

1 ———> MOYENNE

2 ———> LISSAGE MAJORITAIRE

3 ———> TYPE NAGAO (variance min)

4 ———> TYPE NAGAO (variance max)

5 ———> YOKOYA

6 ———> VARIANCE

7 ———> LISSAGE MINORITAIRE

Q5 NOM DE L'IMAGE A CREER ?

Q6 DANS LE DIRECTORY ?

Si le choix en Q4 est égal à 1, 2 ou 7, les questions Q7 et Q8 sont posées.

Q7 NOMBRE D'ITERATIONS ?

Q8 POURCENTAGE ?

Nota : L'arrêt des itérations se produit quand le nombre de pixels modifiés au cours de la dernière itération est égal au pourcentage fixé du nombre de pixels modifiés au cours du tout premier filtrage.

Si le choix en Q4 est égal à 5, les questions Q9 et Q10 sont posées.

Q9 TAILLE DE LA MAILLE ?

Q10 VALEUR DU COEFFICIENT (en réel xxx.xx) ?

PROGRAMME HISBID_COR

FONCTION :

Le programme **HISBID_COR** calcule l'histogramme bidimensionnel sur les variables de contour DIR et PNC, par l'intermédiaire d'un tableau de correspondance provenant du programme TRI-COT. On peut en outre sauvegarder l'histogramme sous forme d'image; le programme gère lui-même le nom du fichier à sauvegarder, en lui attribuant l'extension .HB

APPEL : HISBID_COR

MENU :

- Q1 CHOIX DU NOEUD :
- Q2 NOM DE LA DIRECTORY OU SONT OBLIGATOIREMENT
SITUES LES FICHIERS A TRAITER ?
- Q3 NOM GENERIQUE DU TABLEAU A TRAITER
COMPRENANT TOUTES LES STRUCTURES ?
- Q4 NOM GENERIQUE DU TABLEAU NomFich.COR ISSU DU TRAITEMENT
DU PROGRAMME TRICOT OU BIEN DU PROGRAMME PRESTA ?

Traitement avec affichage sur écran de l'histogramme. En [S]ortie :

- Q5 1 ———> SAUVEGARDE
- 2 ———> SORTIE

PROGRAMME HIST_COR

FONCTION :

Le programme **HIST_COR** calcule l'histogramme sur toutes les variables (variables de contour et de surface). On peut en outre sauvegarder chaque histogramme sous forme d'image, le programme attribuant une extension au nom de la variable choisie (NomFich.RAY, NomFich.DIR, NomFich.CC, etc...).

APPEL : HIST_COR

MENU :

- Q1 CHOIX DU NOEUD :
- Q2 NOM DE LA DIRECTORY OU SONT OBLIGATOIREMENT
SITUES LES FICHIERS A TRAITER (tables et images à lire et à créer) ?
- Q3 NOM GENERIQUE DU TABLEAU A TRAITER
COMPRENANT TOUTES LES STRUCTURES ?
- Q4 NOM GENERIQUE DU TABLEAU NomFich.COR ISSU DU TRAITEMENT
DU PROGRAMME TRICOT OU BIEN DU PROGRAMME PRESTA ?
- Q5 LISTE DES VARIABLES DU TABLEAU NomFich.RES
SUR LESQUELLES ON PEUT INTERVENIR :
- 1 ———> RAYON
 - 2 ———> NOMBRE DE PIXELS
 - 3 ———> POURCENTAGE D'INTERSECTION
 - 4 ———> DIRECTION DE LA NORMALE
 - 5 ———> COEFFICIENT DE CORDE
 - 6 ———> COEFFICIENT DE SYMETRIE
 - 7 ———> PENTE ANGULAIRE
 - 8 à 14 ———> VARIABLES DE SURFACE

CHOIX DE LA VARIABLE ?

Traitement avec affichage sur écran de l'histogramme. En [S]ortie :

- Q6 0 ———> AUTRE CHOIX
1 ———> SAUVEGARDE
2 ———> SORTIE

Si 0 en Q6, retour en Q5 sans Sauvegarde; si 1 en Q6, retour à la question Q5 après Sauvegarde.

PROGRAMME INDICES

FONCTION :

Le programme INDICES calcule, à partir des données SPOT, les indices de végétation, de brillance et de couleur.

APPEL : INDICES

MENU :

Q1 TYPE D'INDICE :

1 ———> INDICE DE BRILLANCE $I_b = [(xs1^2 + xs2^2 + xs3^2)/3]^{1/2}$

2 ———> INDICE DE VEGETATION $I_v = 128[(1 + (xs3 - xs2))/xs3 + xs2]$

3 ———> INDICE DE COULEUR $I_c = 128[(1 + (xs2 - xs1))/xs2 + xs1]$

Q2 CHOIX DU NOEUD :

Q3 DIRECTORY CONTENANT LES CANAUX ?

Si le choix de l'indice est 1, appel des questions Q4, Q5 et Q6.

Si le choix de l'indice est 2, appel à Q5 et Q6.

Si le choix de l'indice est 3, appel à Q4 et Q5.

Q4 NOM DE L'IMAGE XS1 A TRAITER

Q5 NOM DE L'IMAGE XS2 A TRAITER

Q6 NOM DE L'IMAGE XS3 A TRAITER

Q7 NOM DE L'IMAGE A CREER ?

Q8 DANS LA DIRECTORY ?

PROGRAMME LIMITE

FONCTION :

Le programme **LIMITE** issu du logiciel TAPIS produit une carte des sauts de réflectance multi-directionnelle.

APPEL : LIMITE

MENU :

Q1 CHOIX DU NOEUD :

Q2 NOM DU FICHIER A TRAITER ?

Q3 NOM DIRECTORY ?

Q4 IMAGE EN SORTIE
NOM IMAGE ?

Q5 NOM DIRECTORY ?

Q6 DIRECTIONS :
1 ———> NORD-SUD
2 ———> EST-OUEST
3 ———> NE-SW
4 ———> NW-SE
5 ———> Croix grecque
6 ———> Croix de St André
7 ———> Etoile

Nota : 5 correspond à la combinaison 1 + 2; 6 à la combinaison 3 + 4; et 7 à la combinaison 1 + 2 + 3 + 4.

Q7 POSTION DU RESULTAT :
1 ———> val. inf. du couple
2 ———> val. sup. du couple

Nota : lorsque le programme effectue la différence entre les valeurs de deux pixels voisins, il peut mettre la valeur absolue de cette différence sur le plan résultat, soit à l'emplacement de la valeur la plus faible de ces deux pixels, soit à l'emplacement de la valeur la plus forte de ces deux pixels. En cas de combinaison de directions, seule la différence absolue la plus forte est conservée.

PROGRAMME MIN_MAX

FONCTION :

Le programme MIN_MAX calcule la moyenne, la valeur minimum et la valeur maximum pour toutes les variables des classes provenant de l'utilisation des programmes de Statistique.

APPEL : MIN_MAX

MENU :

- Q1 CHOIX DU NOEUD :
- Q2 NOM DE LA DIRECTORY OU SONT OBLIGATOIREMENT
SITUES LES FICHIERS A TRAITER (tables et images à lire et à créer) ?
- Q3 NOM GENERIQUE DES TABLEAUX NomFich.RES ET NomFich.COR
ASSOCIES A NomFich ?
Nota : le nom générique du tableau .COR doit être identique à celui de .RES
- Q4 NOM GENERIQUE DU TABLEAU RESULTAT ?
- Q5 TRAITEMENT SUR LE TABLEAU DE CORRESPONDANCE :
0 ———> OUI 1 ———> NON
- Q6 NOMBRE DE CLASSES ?
- Q7 NOMBRE D'ELEMENTS DANS LA CLASSE n ?
- Q8 NUMERO DE L'ELEMENT m DANS LA CLASSE n ?

Nota : Q7 se répète en fonction de la réponse en Q6, et Q8 en fonction de la réponse en Q7.

PROGRAMME PARPON

FONCTION :

Le programme PARPON (PARTition PONdérée en fonction d'un pourcentage) effectue une partition sur l'image en deux classes dépendant du pourcentage demandé; la première classe est codée 1, la seconde 0.

APPEL : PARPON

MENU :

- Q1 CHOIX DU NOEUD :
- Q2 NOM DU FICHIER A TRAITER ?
- Q3 NOM DIRECTORY ?
- Q4 NOM DE L'IMAGE A CREER ?
- Q5 NOM DIRECTORY ?
- Q6 POURCENTAGE (en integer) ?

PROGRAMME PARSIFAL

FONCTION :

Le programme **PARSIFAL** rassemble en seul traitement les programmes **PARPON**, **FILTRE11**, **LIMITE**, **DEST_ANG** et **DEST_CADRE**. Il engendre, à partir du fichier entrée **NomFich** une image binaire **NomFich.PAR**(pourcentage en integer de la tranche).

APPEL : PARSIFAL

MENU :

- Q1 CHOIX DU NOEUD :**
- Q2 NOM DU FICHIER A TRAITER ?**
- Q3 NOM DIRECTORY ?**
- Q4 POURCENTAGE (en integer) DE LA TRANCHE ?**

PROGRAMME PARSIFAL2

FONCTION :

Le programme **PARSIFAL2** rassemble en seul traitement les programmes **PARPON**, **FILTRES**, **LIMITE**, **DEST_ANG** et **DEST_CADRE**. De plus, il permet d'obtenir en une seule fois, toutes les images correspondant aux différents seuils de réflectance définis par les tranches de pourcentage. Il engendre, à partir du fichier entrée **NomFich** une série d'images binaires **NomFich.PAR**(pourcentage en integer de chacune des tranches). Le nombre d'images créées dépend du seuil en pourcentage défini.

APPEL : PARSIFAL2

MENU :

Q1 CHOIX DU NOEUD :

Q2 NOM DU FICHIER A TRAITER ?

Q3 NOM DIRECTORY ?

Q4 POURCENTAGE (en integer) DES TRANCHES ?

Nota : Pour un seuil de 10%, le programme créera 9 images binaires nommées **NomFich.PAR10**, **NomFich.PAR20**, ..., **NomFich.PAR90**. Seuls les seuils de 1, 2, 5, 10, 20, 25 et 50 sont acceptés par le programme.

PROGRAMME PRELAS_A

FONCTION :

Le programme **PRELAS_A** (**PRE**paration du document pour l'imprimante **LA**Ser) code une image bitmap dont les teintes de gris sont directement assimilables par l'imprimante laser.

APPEL : PRELAS_A

MENU :

- Q1 CHOIX DU NOEUD ?
- Q2 NOM DU FICHIER ?
- Q3 NOM DIRECTORY ?
- Q4 CADRE :
 0 ———> OUI 1 ———> NON

Nota : le programme crée l'image bitmap dans le répertoire /user/NomDir/images/NomFich/bitmap. L'impression sur l'imprimante laser se fait à l'aide de l'ordre : `prf -pr laser bitmap -magn n` (n étant un chiffre qui permet de faire varier la taille de l'image).

PROGRAMME PRESTA

FONCTION :

Le programme **PRESTA** (PREparation des données STATistiques) assure la confection de fichiers transférables et assimilables par le logiciel STATITCF (1990), selon les formats retenus par PAR-ROT et CHEVILLOTTE (1990) dans le logiciel TIMOR. Ce programme peut éliminer les structures dont les valeurs sont supérieures à un seuil donné. Il est également possible de ne pas tenir compte des structures fermées. Le programme crée un tableau de correspondance (NomFich.COR) établissant une correspondance entre le numéro d'ordre des structures extraites et celui qu'ils ont sur le tableau générique issu du programme ADONIS.

APPEL : PRESTA

MENU :

Q1 CHOIX DU NOEUD :

Q2 NOM DE LA DIRECTORY OU SONT OBLIGATOIREMENT
SITUES LES FICHIERS A TRAITER (tables) ?

Q3 NOM DU TABLEAU A TRAITER ?

Nota : il s'agit du fichier NomFich.RES

Q4 NOM GENERIQUE DES TABLEAUX A SAUVEGARDER
Nota : le programme gère lui-même les extensions .VAR, .TRE, .CRE et .COR
Il s'agit des trois fichiers FIVAR, FITREX et FICREX du logiciel STEPPE
et du tableau de correspondance

Q5 NOMBRE DE VARIABLES A EXTRAIRE ?

Q6 NOM DE LA VARIABLE n ?
Nota : utiliser 5 caractères

Q7 LE TABLEAU ETUDIE COMPREND LA SUITE :
1 NUMERO D'ORDRE
2 CENTRE (x)
3 CENTRE (y)
4 RAYON
5 NOMBRE DE PIXELS
6 COEFFICIENT D'INTERSECTION
7 DIRECTION DE LA NORMALE
8 COEFFICIENT DE CORDE
9 COEFFICIENT DE SYMETRIE
10 PENTE ANGULAIRE

- 11 OCCURENCE
12, 13, 14 et 15 EXTREMITES DE LA CORDE
16 NUMERO D'ORDRE
17 et suivant VARIABLES DE SURFACE
- Q8 NUMERO D'ORDRE DE LA VARIABLE n
CORRESPONDANT A NomVar (n) ?
- Q9 ELIMINATION DES STRUCTURES FERMEES :
0 ———> NON 1 ———> OUI
- Q10 ELIMINATION DES VALEURS SUPERIEURES A 999 :
0 ———> NON 1 ———> OUI

PROGRAMME REGIME
PROGRAMME REGIME_VIS

FONCTION :

Le programme **REGIME** produit une squelettisation des formes structurales binaires. Ce module est basé sur les travaux de O'GORMAN, pour une fenêtre glissante de taille 3x3.

APPEL : REGIME

MENU :

Q1 **CHOIX DU NOEUD :**

FICHER ENTREE

Q2 **NOM DU FICHER ?**

Q3 **NOM DU REPERTOIRE ?**

FICHER SORTIE

Q4 **NOM DU FICHER ?**

Q5 **SI LE REPERTOIRE DE SORTIE EST
IDENTIQUE AU REPERTOIRE ENTREE
FRAPPER " = "
SI NON, SAISIR LE NOM DU REPERTOIRE**

NOTA : Le programme **REGIME_VIS** présente le même Menu que **REGIME**. Il permet de plus de suivre en mode graphique le déroulement du programme.

PROGRAMME ROSACE

FONCTION :

Le programme **ROSACE** produit, à partir de la valeur des rayons et de la direction des normales à la corde, deux rosaces de direction, l'une basée sur la plus grande valeur du rayon rencontrée, l'autre sur la fréquence des directions.

APPEL : ROSACE

MENU :

- Q1 **CHOIX DU NOEUD :**
- Q2 **NOM DE LA DIRECTORY OU SONT OBLIGATOIREMENT
SITUES LES FICHIERS A TRAITER (tables et images à lire et à créer) ?**
- Q3 **NOM GENERIQUE DU TABLEAU A TRAITER
COMPRENANT TOUTES LES STRUCTURES ?**
- Q4 **NOM GENERIQUE DU TABLEAU NomFich.COR ISSU DU TRAITEMENT
DU PROGRAMME TRICOT OU BIEN DU PROGRAMME PRESTA ?**

Traitement avec affichage sur écran des rosaces. En [S]ortie :

- Q5 1 ———> SAUVEGARDE
 2 ———> SORTIE

PROGRAMME TRICOT

FONCTION :

Le programme **TRICOT** (**TRI** des **C**ontours **T**raités) trie les structures consignées sur les fichiers **NomFich.RES** et les trace à partir des données du fichier **NomFich.TAB**. Ce tri est effectué à partir des variables de "contour" et/ou de "surface". Ce programme peut également utiliser le fichier **NomFich1.RES** issu du programme **CONCORDE**.

APPEL : TRICOT

MENU :

Q1 CHOIX DU NOEUD :

Q2 NOM DE LA DIRECTORY OU SONT OBLIGATOIREMENT
SITUES LES FICHIERS A TRAITER ?

Q3 NOM GENERIQUE DU TABLEAU A TRAITER ?

Nota : il s'agit du fichier **NomFich.RES**

Q4 NOM GENERIQUE DU TABLEAU ET DE L'IMAGE CORRESPONDANT
AU TABLEAU A OCCURENCE 1 DONT ISSU LE TABLEAU A TRAITER ?

Nota : Le nom de ce tableau peut être identique à celui du tableau générique à traiter. Dans ce cas, frapper "="

Q5 NOM GENERIQUE DE L'IMAGE A CREER ?

LISTE DES VARIABLES DU TABLEAU **NomFich.RES** SUR LESQUELLES ON PEUT
INTERVENIR :

- 1 NOMBRE DE PIXELS
- 2 RAYON
- 3 COEFFICIENT D'INTERSECTION
- 4 DIRECTION DE LA NORMALE
- 5 COEFFICIENT DE CORDE
- 6 COEFFICIENT DE SYMETRIE
- 7 PENTE ANGULAIRE
- 8 OCCURENCE
- 9 et plus : VARIABLES DE SURFACE

Q6 TRAITEMENT SUR LES VARIABLES DE CONTOUR :
0 ———> OUI 1 ———. NON

Nota : si 1 en Q6, passage à Q36.

Q7 PRISE EN COMPTE DU NOMBRE DE PIXELS "
0 ———. OUI 1 ———. NON

Nota : si 1 en Q7, nbpix_min = 1 et nb_pix_max = 1000; passage à Q10.

Q8 VALEUR MINIMUM ?

Q9 VALEUR MAXIMUM ?

Q10 PRISE EN COMPTE DE LA VALEUR DU RAYON :
0 ———> OUI 1 ———> OUI

Nota : si Q10 = 1, ray_min = 1 et ray_max = 1000; passage à Q13.

Q11 VALEUR MINIMUM ?

Q12 VALEUR MAXIMUM ?

Q13 PRISE EN COMPTE DU POURCENTAGE D'INTERSECTION :
0 ———> OUI 1 ———> NON

Nota : si Q13 = 1, CI_min = 20 et CI_max = 100; passage à Q16

Q14 VALEUR MINIMUM ?

Q15 VALEUR MAXIMUM ?

Q16 PRISE EN COMPTE DE LA VALEUR DE DIRECTION :

Nota : si Q16 = 1, DIR_min = 0 et DIR_max = 1000; passage à Q19.

Q17 VALEUR MINIMUM ?

Q18 VALEUR MAXIMUM ?

Q19 PRISE EN COMPTE DU COEFFICIENT DE CORDE :
0 ———> OUI 1 ———> NON

Nota : si Q19 = 1 CC_min = 0 et CC_max = 2000; passage à Q22.

Q20 VALEUR MINIMUM ?

Q21 VALEUR MAXIMUM ?

Q22 PRISE EN COMPTE DE DNC :
0 ———> OUI 1 ———> NON

Nota : si Q22 = 1, DNC_min = 0 et DNC_max = 2000; passage à Q25.

- Q23 VALEUR MINIMUM ?
- Q24 VALEUR MAXIMUM ?
- Q25 PRISE EN COMPTE DE LA PENTE :
0 ———> OUI 1 ———> NON

Nota : si Q25 = 1, PNC_min = -90 et PNC_max = 2000; passage à Q33.

- Q26 1 ———> UN INTERVALLE
2 ———> DEUX INTERVALLES

Nota : si Q26 = 1, Q27 et Q28; si Q26 =2, Q29 à Q32.

- Q27 VALEUR MINIMUM ?
- Q28 VALEUR MAXIMUM ?
- puis, passage à Q33.

- Q29 PREMIER INTERVALLE :
VALEUR MINIMUM ?

- Q30 VALEUR MAXIMUM ?

- Q31 DEUXIEME INTERVALLE :
VALEUR MINIMUM ?

- Q32 VALEUR MAXIMUM ?

- Q33 PRISE EN COMPTE DU SEUIL D'OCCURENCE :
0 ———> OUI 1 ———> NON

Nota : si 1 en Q33, OCC_min = 1 et OCC_max = 1000; passage à Q36.

- Q34 VALEUR MINIMUM ?

- Q35 VALEUR MAXIMUM ?

- Q36 TRAITEMENT SUR LES VARIABLES DE SURFACE ?
Nota : s'il s'agit d'un fichier provenant du programme CONCERT_VIS.
0 ———> OUI 1 ———> NON

Nota : si Q36 = 1, passage à Q41.

- Q37 NOMBRE DE VARIABLES DE CE FICHIER ?

- Q38 PRISE EN COMPTE DE LA VARIABLE n ?
0 ———> OUI 1 ———> NON

Q39 (si 0 en Q38)

VALEUR MINIMUM ?

Q40 (si 0 en Q38)

VALEUR MAXIMUM ?

Nota : les questions Q39 et Q40 sont posées n fois, en fonction de la réponse faite en Q37.

Q41

IL EST DE PLUS POSSIBLE DE :

(A) TRACER LES CONTOURS DETECTES AVEC OU SANS CORDE ET NORMALE

(B) TRACER LES CERCLES CORRESPONDANTS AUX STRUCTURES DETECTEES

INDIQUEZ VOTRE CHOIX :

1 ———> TRACER LES CONTOURS DETECTES

2 ———> TRACER LES CERCLES

Nota : si 1 en Q41, question Q42; si 2 en Q41, question Q43.

Q42

1 ———> CONTOURS DETECTES

2 ———> CONTOURS (+ CORDE ET NORMALE)

Q43

[P]érimètre

[S]urface

IV.- MODULES UTILITAIRES

Nombre de modules utilitaires figurent dans la librairie TAPIS_LIB, mais n'ont pas été documentés dans la notice relative au logiciel TAPIS (PARROT op. cit.). Par ailleurs, de nouveaux utilitaires ont été développés à l'occasion de la mise au point du logiciel ADONIS; ces nouveaux utilitaires ont été placés dans la librairie ADONIS_LIB.

Tous ces utilitaires sont décrits dans les pages qui suivent, [A] et [T] indiquant qu'ils sont respectivement issus des librairies ADONIS_LIB et TAPIS_LIB. Toutefois, tous les modules décrits ici ont été regroupés dans la librairie ADONIS_LIB. Il n'est présenté ici que de modules gérant les entrées-sorties.

IV.A.- TYPE_NODE [T]

Fonction :

Ce module permet d'aller chercher l'image où elle se trouve, quelque soit le noeud du réseau utilisé.

Appel :

call type_node(nom_node)

Arguments :

(a) nom_node : noeud du réseau où se situe l'image.

Déclarations :

character*32 nom_node

Menu :

CHOIX DU NOEUD DU RESEAU

- 1 ———> 5141
- 2 ———> DF91
- 3 ———> DFFE
- 4 ———> SOLEIL
- 5 ———> MARS
- 6 ———> VENUS
- 7 ———> KRONOS

IV.B.– FICH_ENTREE [A]

Fonction :

Saisie du nom du fichier et du répertoire où se situe le fichier à traiter. En sortie, cheminement (pathname) de l'image et de son descriptif, ainsi que le nom du répertoire en vue de son utilisation par le module FICH_SORTIE.

Appel :

call fich_entree(nom_node,nomim_in,nomdescr_in,nomrep)

Arguments :

- (a) nom_node : noeud du réseau où se situe l'image.
- (b) nomim_in : cheminement (pathname) de l'image.
- (c) nomdescr_in : cheminement du descriptif.
- (d) nomrep : nom du répertoire (défini à l'intérieur du module).

Déclarations :

character*32 nom_node,nomrep
character*256 nomim_in,nomdescr_in

Menu :

FICHIER ENTREE

Q1 NOM DU FICHIER ?

Q2 NOM DU REPERTOIRE ?

IV.C.– FICH_ENTREE_NF [A]

Fonction :

Même fonction que FICH_ENTREE, mais de plus renvoie le nom du fichier entrée, défini à l'intérieur du module, en vue d'une utilisation éventuelle par l'utilitaire FICH_SORTIE_NF.

Appel :

call fich_entree_nf(nom_node,nomim_in,nomdescr_in,nomrep,nomfich_in)

Arguments :

- (a) nom_node : noeud du réseau où se situe l'image.
- (b) nomim_in : cheminement (pathname) de l'image.
- (c) nomdescr_in : cheminement du descriptif.
- (d) nomrep : nom du répertoire (défini à l'intérieur du module).

(e) nomfich_in : nom du fichier entrée (défini à l'intérieur du module).

Déclarations :

```
character*32 nom_node,nomrep,nomfich_in  
character*256 nomim_in,nomdescr_in
```

Menu :

FICHIER ENTREE

Q1 NOM DU FICHIER ?

Q2 NOM DU REPERTOIRE ?

IV.D.- FICH_ENTREE2 [A]

Fonction :

Le module FICH_ENTREE2 saisit le nom du fichier entrée et de son répertoire, ouvre le descriptif, le lit, si besoin le complète en faisant appel à l'utilitaire VERCOR2, et le ferme, renvoyant les nombres de lignes et de colonnes de l'image, le type de l'image qui est confronté au type demandé, la dynamique donnée par le descriptif et la taille des pixels de l'image. Il ouvre le fichier image sur l'unité logique spécifiée par le programme principal et renvoie le nom du répertoire utilisable par l'ensemble des modules de sortie (FICH_SORTIE, FICH_SORTIE_NF, FICH_SORTIE2, FICH_SORTIE2_NF).

Appel :

```
call fich_entree2(nom_node,unit_im_in,nb_lig,nb_col,type,min,max,  
+               taille_pix,type_dem,nomrep)
```

Arguments :

- (a) nom_node : noeud du réseau où se situe l'image.
- (b) unit_im_in : numéro de l'unité logique où le fichier image doit être ouvert.
- (c) nb_lig : nombre de lignes.
- (d) nb_col : nombre de colonnes.
- (e) type : type de l'image (1 pour 8 bits; 4 pour integer 4).
- (f) min : minimum de la dynamique.
- (g) max : maximum de la dynamique.
- (h) taille_pix : taille de l'arête du pixel carré (en décimètres).
- (i) type_dem : type de l'image que traite le programme principal (1 ou 4).
- (j) nomrep : nom du répertoire (défini à l'intérieur du module).

Déclarations :

```
character*32 nom_node,nomrep  
integer*4 unit_im_in,nb_lig,nb_col,type,min,max,taille_pix,type_dem
```

Menu :

FICHER ENTREE

Q1 NOM DU FICHER ?

Q2 NOM DU REPERTOIRE ?

Nota : si le descriptif est incomplet, l'appel au module VERCOR2 entraîne les questions figurant dans ce module (cf. plus loin). Une absence de correspondance entre le type de l'image et le type demandé provoque une interruption du programme.

IV.E.- FICH_ENTREE2_NF [A]**Fonction :**

Identique à celle de l'utilitaire FICH_ENTREE2, mais permet de ne pas avoir à réentrer le nom du fichier image défini en entrée.

Appel :

```
call fich_entree2(nom_node,unit_im_in,nb_lig,nb_col,type,min,max,  
+               taille_pix,type_dem,nomrep,nomfich_in)
```

Arguments :

- (a) nom_node : noeud du réseau où se situe l'image.
- (b) unit_im_in : numéro de l'unité logique où le fichier image doit être ouvert.
- (c) nb_lig : nombre de lignes.
- (d) nb_col : nombre de colonnes.
- (e) type : type de l'image (1 pour 8 bits; 4 pour integer 4).
- (f) min : minimum de la dynamique.
- (g) max : maximum de la dynamique.
- (h) taille_pix : taille de l'arête du pixel carré (en décimètres).
- (i) type_dem : type de l'image que traite le programme principal (1 ou 4).
- (j) nomrep : nom du répertoire (défini à l'intérieur du module).
- (k) nomfich_in : nom du fichier image (défini à l'intérieur du module).

Déclarations :

```
character*32 nom_node,nomrep,nomfich_in  
integer*4 unit_im_in,nb_lig,nb_col,type,min,max,taille_pix,type_dem
```

Menu :

FICHER ENTREE

Q1 NOM DU FICHER ?

Q2 NOM DU REPERTOIRE ?

Nota : si le descriptif est incomplet, l'appel au module VERCOR2 entraîne les questions figurant dans ce module (cf. plus loin). Une absence de correspondance entre le type de l'image et le type demandé provoque une interruption du programme.

IV.F.- FICH_SORTIE [A]

Fonction :

Saisie du nom du fichier et du répertoire du fichier sortie. En sortie, cheminement (pathname) de l'image et de son descriptif.

Appel :

call fich_sortie(nom_node,nomim_out,nomdescr_out,nomrep)

Arguments :

- (a) nom_node : noeud du réseau où se situe l'image.
- (b) nomim_out : cheminement (pathname) de l'image de sortie.
- (c) nomdescr_out : cheminement du descriptif de l'image de sortie.
- (d) nomrep : nom du répertoire (défini à l'intérieur de l'un des modules d'entrée).

Déclarations :

character*32 nom_node,nomrep
character*256 nomim_out,nomdescr_out

Menu :

FICHER SORTIE

Q1 NOM DU FICHER ?

Q2 SI LE REPERTOIRE DE SORTIE EST
IDENTIQUE AU REPERTOIRE ENTREE
FRAPPER " = "
SI NON, SAISIR LE NOM DU REPERTOIRE

IV.G.– FICH_SORTIE_NF [A]

Fonction :

Même fonction que FICH_SORTIE, mais de plus, cet utilitaire permet de ne pas avoir à modifier le nom du fichier saisi en entrée.

Appel :

call fich_sortie_nf(nom_node,nomim_out,nomdescr_out,nomrep,nomfich_in)

Arguments :

- (a) nom_node : noeud du réseau où se situe l'image.
- (b) nomim_out : cheminement (pathname) de l'image.
- (c) nomdescr_out : cheminement du descriptif.
- (d) nomrep : nom du répertoire (défini à l'intérieur d'un module entrée).
- (e) nomfich_in : nom du fichier entrée (défini à l'intérieur de l'un des modules entrée).

Déclarations :

character*32 nom_node,nomrep,nomfich_in
character*256 nomim_out,nomdescr_out

Menu :

FICHIER SORTIE

- | | |
|----|--|
| Q1 | SI LE FICHIER DE SORTIE EST
IDENTIQUE AU FICHIER ENTREE
FRAPPER " = "
SI NON, SAISIR LE NOM DU FICHIER |
| Q2 | SI LE REPERTOIRE DE SORTIE EST
IDENTIQUE AU REPERTOIRE ENTREE
FRAPPER " = "
SI NON, SAISIR LE NOM DU REPERTOIRE |

IV.H.– FICH_SORTIE2 [A]

Fonction :

Le module FICH_SORTIE2 saisit le nom du fichier sortie et de son répertoire, teste si la place disponible sur le disque est suffisante (appel du module RESERVE_DISP), ouvre l'unité logique du descriptif, y consigne le nombre de lignes et de colonnes et le type de l'image sortie, et ouvre l'unité logique de l'image.

Appel :

call fich_sortie2(nom_node,unit_im_out,unit_descr_out,nb_lig,nb_col,type_sortie,
+ nomrep,nb_oct_dem)

Arguments :

- (a) nom_node : noeud du réseau où se situe l'image.
- (b) unit_im_out : numéro de l'unité logique où le fichier image doit être ouvert.
- (c) unit_descr_out : numéro de l'unité logique où le fichier descriptif doit être ouvert.
- (d) nb_lig : nombre de lignes.
- (e) nb_col : nombre de colonnes.
- (f) type_sortie : type de l'image (1 pour 8 bits; 4 pour integer 4).
- (g) nomrep : nom du répertoire (défini à l'intérieur de l'un des modules entrée).
- (h) nb_oct_dem : nombre d'octets nécessaires pour consigner l'image.

Déclarations :

character*32 nom_node,nomrep
integer*4 unit_im_out,unit_descr_out,nb_lig,nb_col,type_sortie,nb_oct_dem

Menu :**FICHIER SORTIE**

- | | |
|----|--|
| Q1 | NOM DU FICHIER ? |
| Q2 | SI LE REPERTOIRE DE SORTIE EST
IDENTIQUE AU REPERTOIRE ENTREE
FRAPPER " = "
SI NON, SAISIR LE NOM DU REPERTOIRE |
| Q2 | NOM DU REPERTOIRE ? |

IV.L- FICH_SORTIE2_NF [A]**Fonction :**

Identique à celle de l'utilitaire FICH_SORTIE2, mais permet de ne pas avoir à réentrer le nom du fichier image défini en entrée.

Appel :

call fich_sortie2(nom_node,unit_im_out,unit_descr_out,nb_lig,nb_col,type_sortie,
+ nomrep,nomfich_in,nb_oct_dem)

Arguments :

- (a) nom_node : noeud du réseau où se situe l'image.
- (b) unit_im_out : numéro de l'unité logique où le fichier image doit être ouvert.

- (c) unit_descr_out : numéro de l'unité logique où le fichier descriptif doit être ouvert.
- (d) nb_lig : nombre de lignes.
- (e) nb_col : nombre de colonnes.
- (f) type_sortie : type de l'image (1 pour 8 bits; 4 pour integer 4).
- (g) nomrep : nom du répertoire (défini à l'intérieur de l'un des modules entrée).
- (h) nomfich_in : nom du fichier entrée.
- (i) nb_oct_dem : nombre d'octets nécessaires pour consigner l'image.

Déclarations :

```
character*32 nom_node,nomrep,nomfich_in
integer*4 unit_im_out,unit_descr_out,nb_lig,nb_col,type_sortie,nb_oct_dem
```

Menu :

FICHIER SORTIE

- Q1 SI LE FICHIER DE SORTIE EST
 IDENTIQUE AU FICHIER ENTREE
 FRAPPER " = "
 SI NON, SAISIR LE NOM DU FICHIER
- Q2 SI LE REPERTOIRE DE SORTIE EST
 IDENTIQUE AU REPERTOIRE ENTREE
 FRAPPER " = "
 SI NON, SAISIR LE NOM DU REPERTOIRE

IV.J.- FICH_ENTREE_SORTIE [A]

Fonction :

Ce module gère les entrées et les sorties, regroupant les fonctionnalités de FICH_ENTREE2_NF et FICH_SORTIE2_NF.

Nota :

L'appel au module TYPE_NODE est effectué directement par l'utilitaire FICH_ENTREE_SORTIE.

Appel :

```
+           call fich_entree_sortie(nom_node,unit_im_in,unit_im_out,unit_descr_out,type_entree,
                                  type_sortie,nb_lig,nb_col,taille_pix,nb_oct_dem)
```

Arguments :

- (a) nom_node : noeud du réseau où se situe l'image.
- (b) unit_im_in : numéro de l'unité logique où le fichier entrée doit être ouvert.
- (b) unit_im_out : numéro de l'unité logique où le fichier sortie doit être ouvert.
- (c) unit_descr_out : numéro de l'unité logique où le fichier descriptif de sortie doit être

ouvert.

(d) type_entree : type de l'image en entrée (1 pour 8 bits; 4 pour integer 4).

(e) type_sortie : type de l'image en sortie (1 pour 8 bits; 4 pour integer 4).

(f) nb_lig : nombre de lignes.

(g) nb_col : nombre de colonnes.

(h) taille_pix : taille du pixel.

(i) nb_oct_dem : nombre d'octets nécessaires pour consigner l'image.

Déclarations :

```
character*32 nom_node
integer*4 unit_im_in,unit_im_out,unit_descr_out,nb_lig,nb_col,taille_pix
integer*4 type_entree,type_sortie,nb_oct_dem
```

Menu :

FICHER ENTREE

Q1 NOM DU FICHER ?

Q2 NOM DU REPERTOIRE ?

FICHER SORTIE

Q3 SI LE FICHER DE SORTIE EST
IDENTIQUE AU FICHER ENTREE
FRAPPER " = "
SI NON, SAISIR LE NOM DU FICHER

Q4 SI LE REPERTOIRE DE SORTIE EST
IDENTIQUE AU REPERTOIRE ENTREE
FRAPPER " = "
SI NON, SAISIR LE NOM DU REPERTOIRE

IV.K.- FICH_ENTREE_SORTIE_EXT [A]

Fonction :

Cet utilitaire gère les entrées et les sorties, donnant à l'image en sortie le nom de l'image entrée plus une extension.

Nota :

L'appel au module TYPE_NODE est effectué directement par l'utilitaire FICH_ENTREE_SORTIE_EXT.

Appel :

+ call fich_entree_sortie(nom_node,unit_im_in,unit_im_out,unit_descr_out,type_entree,
type_sortie,nb_lig,nb_col,taille_pix,ext,nb_oct_dem)

Arguments :

- (a) nom_node : noeud du réseau où se situe l'image.
- (b) unit_im_in : numéro de l'unité logique où le fichier entrée doit être ouvert.
- (b) unit_im_out : numéro de l'unité logique où le fichier sortie doit être ouvert.
- (c) unit_descr_out : numéro de l'unité logique où le fichier descriptif de sortie doit être ouvert.
- (d) type_entree : type de l'image en entrée (1 pour 8 bits; 4 pour integer 4).
- (e) type_sortie : type de l'image en sortie (1 pour 8 bits; 4 pour integer 4).
- (f) nb_lig : nombre de lignes.
- (g) nb_col : nombre de colonnes.
- (h) taille_pix : taille du pixel.
- (i) ext : nom de l'extension (par exemple : tab)
- (j) nb_oct_dem : nombre d'octets nécessaires pour consigner l'image.

Déclarations :

```
character*32 nom_node,ext
integer*4 unit_im_in,unit_im_out,unit_descr_out,nb_lig,nb_col,taille_pix
integer*4 type_entree,type_sortie,nb_oct_dem
```

Menu :**FICHER ENTREE**

Q1 NOM DU FICHER ?

Q2 NOM DU REPERTOIRE ?

LE FICHER SORTIE S'APPELERA :
NomFich_entree.EXT

IV.L.- RESERVE_DISP [A]**Fonction :**

Cet module permet de savoir si l'on dispose de place sur le disque pour lancer le programme.
Son usage doit être associé à celui du module FINAL qui purge le fichier //nom_node/tapis/reserve_node.

Appel :

call reserve_disp(nom_node, nb_oct_dem)

Variables externes (situées dans le programme principal) :

- (a) `nom_node` : correspond au nom du noeud du réseau sur lequel sont situés les fichiers et les images traités. `nom_node` est fourni par l'utilisation de l'utilitaire `TYPE_NODE`.
- (b) `nb_oct_dem` : il s'agit du nombre d'octets que réclame la mise en oeuvre du programme; il correspond généralement à `nb_lig*nb_col`.

Variables internes (situées dans le module) :

- (1) `nb_oct_res` : il s'agit du nombre d'octets réservés par le ou les utilisateurs précédents. Le contenu de cette variable est lu dans un fichier "reserve_node" situé ou créé s'il n'existe pas, dans le répertoire `//nom_noeud/tapis`. On évite ainsi de lancer le programme s'il y a de la place disponible dans le noeud, mais si cette place est en partie réservée par un autre utilisateur.
- (2) `nb_oct` : c'est le nombre d'octets disponibles indiqué par la commande "lvofs".
- (3) `nb_oct_disp` : est égal à `nb_oct - nb_oct_res`.

Nota :

- (a) si `nb_oct_dem` est plus grand que `nb_oct_disp`, le module affiche un message indiquant qu'il est impossible d'exécuter le programme. Si la place est suffisante, `nb_oct_dem` est additionné à `nb_oct_res` dans le fichier `"//nom_noeud/tapis/reserve_node"`.
- (b) **ATTENTION**, il ne faut pas arrêter le programme (CTL Q) afin de ne pas fausser le contenu du fichier de contrôle `"//nom_noeud/tapis/reserve_node"`. En cas de fausse manoeuvre, penser à purger ou détruire ce fichier.

Déclarations et programmation dans le programme appelant :

```
integer*4 nb_oct_dem, nb_lig, nb_col
character nom_node*32

nb_oct_dem = nb_lig*nb_col
call reserve_disp(nom_node, nb_oct_dem)
call traitement()
call final(nom_node, nb_oct_dem)
end
```

IV.M.- FINAL [A]**Fonction :**

Cet utilitaire purge le fichier `"//nom_noeud/tapis/reserve_node"`, après déroulement du programme principal.

Appel :

```
call final(nom_node, nb_oct_dem)
```


Arguments :

- (a) nom_node : noeud du réseau où se situe l'image.
- (b) nb_oct_dem : nombre d'octets nécessaires pour consigner l'image.

Déclarations :

```
character*32 nom_node  
integer*4 nb_oct_dem
```

IV.N.- VERCOR2 [A]

Fonction :

Ce module lit le descriptif de l'image entrée, le complète si ce descriptif est incomplet, et renvoie le nombre de lignes et de colonnes, le type, les minimum et les maximum de la dynamique et la taille de l'arête du pixel en centimètres.

Appel :

```
call vercor2(nomim_in,nomdescr_in,nb_lig,nb_col,type,min,max,taille_pix)
```

Arguments :

- (a) nomim_in : cheminement de l'image.
- (b) nomdescr_in : cheminement du descriptif.
- (c) nb_lig : nombre de lignes.
- (d) nb_col : nombre de colonnes.
- (e) type : type de l'image (1 pour 8 bits; 4 pour integer 4).
- (f) min : minimum de la dynamique de l'image.
- (g) max : maximum de la dynamique de l'image.
- (h) taille_pix : taille de l'arête du pixel.

Déclarations :

```
character*256 nomim_in,nomdescr_in  
integer*4 nb_lig,nb_col,type,min,max,taille_pix
```

Menu :

Une ou deux questions sont posées selon l'absence d'informations enregistrés.

TYPE DE L'IMAGE (1 = 8 BITS; 4 = INTEGER4) ?

TAILLE DU PIXEL (en decimetres) ?

IV.O.– CON_BIN [A]

Nota :

Il existe 3 versions du module CON_BIN, selon la taille de l'image traitée : CON_BIN2000, CON_BIN4000 et CON_BIN6000.

Fonction :

Cet utilitaire vérifie si l'image entrée est ou n'est pas de type binaire, et renvoie un message.

Appel :

call con_bin....(nb_lig,nb_col,unit_im_in)

Arguments :

- (a) nb_lig : nombre de lignes.
- (b) nb_col : nombre de colonnes.
- (c) unit_im_in : numéro de l'unité logique où se situe l'image.

Déclarations : (par exemple, pour con_bin2000)

```
character*1 tab(2000,2000)
integer*4 nb_lig,nb_col,unit_im_in

common /E1/tab
```

IV.P.– LECTURE [A]

Fonction :

Lecture de l'image d'entrée.

Nota :

A l'instar de CON_BIN, il existe trois versions de ce module.

Appel :

call lecture....(nb_lig,nb_col,unit_im_in)

Arguments :

- (a) nb_lig : nombre de lignes.
- (b) nb_col : nombre de colonnes.
- (c) unit_im_in : numéro de l'unité logique où se situe l'image.

Déclarations : (par exemple, pour lecture2000)

```
character*1 tab(2000,2000)
```

integer*4 nb_lig,nb_col,unit_im_in

common /E1/tab

IV.Q.- REMARQUES

Les modules utilitaires provenant de l'ensemble UTIL1 (sous /perso/jyb/prog), tels que ECR_DISK_DIRECT, LEC_DISK_DIRECT, LONG_NOM et ERREUR, modules couramment utilisés, ont été placés dans ADONIS_LIB pour des raisons de commodités.

IV.R.- ORGANIGRAMME

L'organigramme de la page suivante illustre l'utilisation qui peut être faite des différents modules gérant les entrées-sorties.

