

MESRS

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT D'ELECTRONIQUE

C.S.T.N. LABORATOIRE DE TELEDETECTION

PROJET DE FIN D'ETUDES
OPTION INGENIORAT EN ELECTRONIQUE

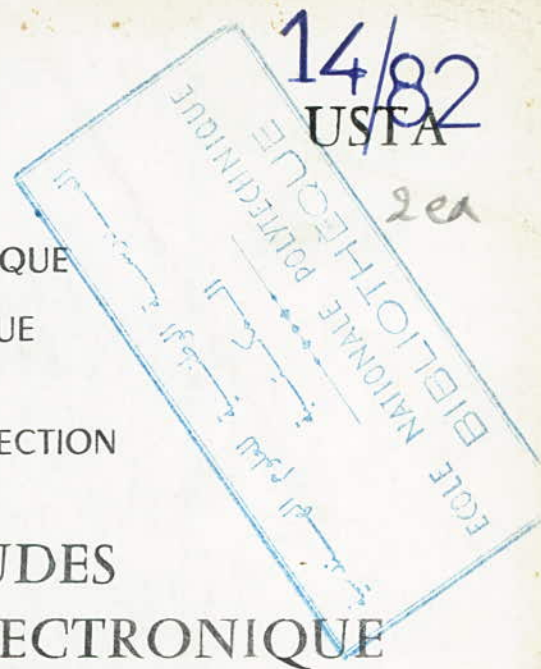
RECONNAISSANCE DE FORMES
ANALYSE DE TEXTURE
EN TRAITEMENT NUMERIQUE D'IMAGES

Proposé par :
A. ABDELLAOUI
A. OUSSEDIK



Etudie par :
A. BENMERZOUGA
R. DRAI

JUIN 82



DE D I C A C E S

A mes parents.
A mes frères.
A mes soeurs.
A mes amis.
A toute ma famille.

ALL

A la mémoire de ma mère.
A la mémoire de mon frère.
A mon père.
A ma tante.
A mes frères et soeurs.
A toute ma famille.
A tous mes amis.

REDOUANE

DE D I C A C E S



REMERCIEMENTS

Nous tenons à remercier notre promoteur M^r.ABDELLAOUI directeur du groupe de recherche en TELEDETECTION, de nous avoir accueillis dans son laboratoire pour l'élaboration de notre projet.

Nos vifs remerciements vont également à M^r.BELKACEMI, M^r.OUSSEDIK et M^{elle}.TILIKETE qui n'ont pas hésité à nous aider tout au long de notre travail.

Nous tenons aussi à remercier G.BAHIA, S.FATIHA et M.KAMAL pour leur aide pratique et morale.

Nous n'oublions pas de remercier le groupe du centre de calcul et particulièrement à M^r.BENDIFALLAH.

Ainsi que M^r.AGOUNI.

Enfin, nous remercions M^r.CHERFA ALI pour sa contribution à la réalisation matérielle de cet ouvrage.

Que tous ceux qui, ont contribué de près ou de loin à l'élaboration de ce projet trouvent ici l'expression de notre profonde reconnaissance.

REMERCIEMENTS

INTRODUCTION

PREMIERE PARTIE

Généralités

1. Définitions:

-IMAGE

-NIVEAU DE GRIS

-IMAGE MULTISPECTRALE, SIGNATURE SPECTRALE

-SIGNATURE SPATIALE

-IMAGE ECHANTILLONNEE

-PIXEL

-RESOLUTION SPATIALE

-CLASSE

-REGLE DE DECISION

-CLASSIFICATION MULTIDIMENSIONNELLE

2. Acquisition de données

a-LES CAPTEURS

b-LES PLATES-FORMES D'OBSERVATION

3. Les satellites LANDSAT

3.1. HISTORIQUE

3.2. RESOLUTIONS SPECTRALES

a-LES CAMERAS R.B.V.

b-LE M.S.S.

b.1. Résolution spectrale des M.S.S.

b.2. Caractéristiques des différentes bandes spectrales

DEUXIEME PARTIE

Processus de reconnaissance de formes

1. INTRODUCTION ET DEFINITIONS

2. ACQUISITION ET TRAITEMENT D'INFORMATION

2.1. ACQUISITION DE L'IMAGE DE LA SCENE

2.2. LA SEGMENTATION

2.2.1. DETECTION DE CONTOURS

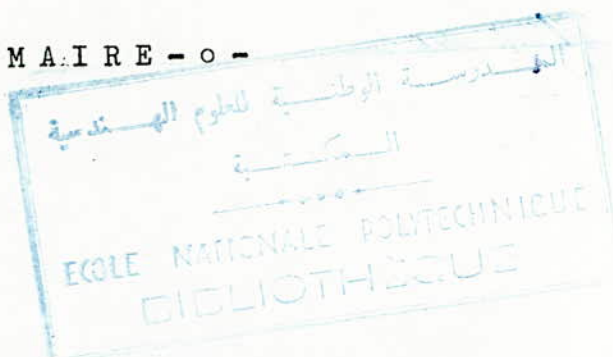
2.2.2. RECHERCHE DU CONTOUR DES FORMES

3. METHODES

3.1. METHODE DU GRADIENT

3.2. METHODE DE HUECKEL

3.3. METHODE A BASE STATISTIQUE



- 3.4.UTILISATION DE LA METHODE DU GRADIENT
 - a-Gradient SIMPLE
 - b-Gradient AMELIORE
 - c-Tracé d'HISTOGRAMME DE gradient
- 4.Affinage et lissage des lignes de contour
 - 4.1.FILTRAGE ANALYTIQUE
 - 4.2.LISSAGE DES LIGNES DE CONTOUR
- 5.Traitement d'une image photographique aérienne digitalisée
 - 5.1.OBTENTION DE L'IMAGE NUMERIQUE
 - 5.2.ENREGISTREMENT DES DONNEES
 - 5.3.TRANSFORMATION DE L'IMAGE EN UNE IMAGE GRADIENT
 - a-Gradient simple
 - b-Gradient amélioré
 - c-Comparaison
 - 5.4.SONCEPTION DU PROGRAMME DE DETECTION DE CONTOURS

TROISIEME PARTIE

- 1.TEXTURE DES IMAGES DE TELEDETECTION
 - 1.1.DEFINITION DE LA TEXTURE
 - 1.2.PLACE DE LA TEXTURE DANS L'INTERPRETATION D'IMAGES DE T.D.
 - 1.3.TECHNIQUES D'ANALYSE DE LA TEXTURE
- 2.LES TRANSFORMATIONS LOCALES
 - 2.1.INTERET DES TRANSFORMATIONS LOCALES POUR L'ETUDE DE LA TEXTURE
 - 2.2.TRANSFORMATIONS LOCALES -DEFINITION,THEORIE
 - 2.2.1.IMAGE DIGITALE
 - 2.2.2.VOISINAGE
 - 2.2.3.TRANSFORMATION LOCALE
 - 2.2.3.1.DEFINITION
 - 2.2.3.2.INVARIANCE EN POSITION
 - 2.2.3.3.TRANSFORMATION LOCALE LINEAIRE
 - 2.2.3.4.DOMAINE DE L'IMAGE TRANSFORMEE
 - 2.3.DESCRPTION DES TRANSFORMATIONS LOCALES UTILISEES
 - 2.3.1.MOYENNE
 - 2.3.2.ECART A LA MOYENNE
 - 2.3.3.ECART TYPE
 - 2.3.4.AUGMENTATION DU CONTRASTE
 - 2.4.PRESENTATION DES RESULTATS
 - 2.4.1.ORGANIGRAMMES ET PROGRAMMES
 - 2.4.2.COMMENTAIRES

00--
I
I
SELON L'ENCYCLOPEDIA UNIVERSALIS:

"La TELEDETECTION est un ensemble de technique mises en oeuvre à partir d'avions, de ballons, de satellites et qui ont pour but d'étudier soit la surface de la terre ou d'autres planètes, soit l'atmosphère, en utilisant les propriétés des ondes électromagnétiques émises, réfléchies ou diffractées par les différents corps observés."

UNE OPERATION DE TELEDETECTION S'EFFECTUE EN TROIS ETAPES

I--ACQUISITION DES DONNEES

II--TRAITEMENT DES DONNEES

III--INTERPRETATION.

I
00--
I

-o- I N T R O D U C T I O N -o-

Depuis de nombreuses années, l'utilisation des photographies aériennes constitue une méthode de couverture rapide et globale de la surface terrestre. Des photo-interprètes travaillant dans divers domaines d'application comme l'agriculture, la géographie, la géologie, l'hydrologie... sont capables de traduire les images reçues en données utiles concernant les objectifs recherchés.

La TELEDETECTION aérospatiale (extension et amélioration notable des techniques de photographie aérienne) conduit à un accroissement considérable du volume des données: une seule image du satellite LANDSAT contient 7.581.600 points dans chacune des quatre bandes spectrales et le satellite peut acquérir plusieurs dizaines d'images par jour. Un nombre important d'information nécessite l'usage d'un traitement par ordinateur.

Les premières techniques de traitement d'images de teledetection sont surtout des techniques multispectrales, chaque point de l'image est affecté de plusieurs paramètres représentant chacun une grandeur liée à l'énergie reçue par le capteur dans une bande spectrale donnée.

Ces traitements considèrent le point de l'image comme isolé: ils ne tiennent pas compte des relations entre le ton de gris du point et celui de ses voisins. Or, une grande partie de l'information de l'image est contenue dans ces relations spatiales.

L'objet de notre travail est l'étude de la notion de texture et de reconnaissance de forme dans l'analyse d'images; la conception d'un logiciel et son application à une image contenant la région d'ALGER.

La première partie de notre mémoire consiste à donner des définitions concernant la télédétection.

La deuxième partie fait l'objet d'application de quelques algorithmes simples de reconnaissance de formes et de détection de contours sur une image photographique aérienne digitalisée.

Enfin dans la troisième partie, nous introduisons la notion de texture et son application sur des images LANDSAT.

P R E M I E R E

P A R T I E

GENERALITES

Les images aérospatiales peuvent se présenter sous deux formes différentes, photographiques ou numériques suivant leurs origines.

Par rapport à l'image photographique usuelle, qui nous permet d'avoir des informations sur les géométries et les couleurs des objets, l'image numérique (généralement portée sur un support magnétique) nous offre en plus l'avantage de posséder des informations sur les propriétés physiques de ces objets. Néanmoins, ces deux types d'images peuvent être traitées et traduites de la même manière.

1. DEFINITIONS:

-IMAGE:

Une image peut être considérée comme une fonction $I(x,y)$ à deux variables réelles et à valeur réelle qui vaut zéro à l'extérieur d'un domaine D et a une valeur bornée positive ou nulle à l'intérieur de ce domaine.

$$\begin{aligned} 0 \leq I(x,y) \leq M \quad \forall (x,y) \in D \\ I(x,y) = 0 \quad \forall (x,y) \notin D \end{aligned}$$

-NIVEAU DE GRIS:

La valeur de la fonction $I(x,y)$ en un point donné est appelée niveau de gris de l'image au point (x,y) ; on trouve encore comme dénomination luminance, intensité en un point, ou radiance.

-IMAGE MULTISPECTRALE, SIGNATURE SPECTRALE:

Dans une image multispectrale, à un point (x,y) de l'image correspond N valeurs correspondant aux N bandes spectrales d'analyse. Ces N valeurs sont appelées signature spectrale au point (x,y) .

-SIGNATURE SPATIALE:

Dans une image, à un point (x,y) on associe M paramètres spatiaux tenant compte du voisinage de celui-ci. Ces M valeurs sont appelées signature spatiale.

-IMAGE ECHANTILLONNEE:

Les images sur lesquelles nous travaillons correspondent à un échantillonnage de la fonction image précédente. L'échantillonnage est réalisé à la fois sur la position (x,y) et

sur l'intensité au point (x,y) .

Ainsi une image monospectrale de taille $LONGL \times LONGC$ à K niveaux de gris est une image dont le nombre total de point est fini

$(LONGL \times LONGC)$ et dont le niveau de gris en un quelconque de ces points est un entier compris entre 0 et $(K-1)$.

Une image $LONGL \times LONGC$ à K niveau de gris est donc une matrice à $LONGL$ lignes et $LONGC$ colonnes à valeurs entières positives ou nulles comprises entre 0 et $(K-1)$

		0	1	2...	j...	(LONGC-1)
0		
1		
2		
⋮	⋮	⋮	⋮	⋮		
i
⋮	⋮	⋮	⋮	⋮		
LONGL-1		

-PIXEL:

Signifiant point elementaire, c'est un element de la matrice precedente.

-RESOLUTION SPATIALE:

La resolution spatiale d'une image numérique échantillonnée est la surface au sol que représente un pixel.

Pour un radiometre à balayage, le terrain est analysé bande par bande. L'angle de balayage (θ) et la longueur (l) pris comme échantillon dans la bande de balayage, fixent la resolution spatiale.

Exemple: La resolution spatiale du capteur MSS du satellite

LANDSAT est de $57m \times 79m = 4500m^2$

-CLASSE:

Une classe dans une image numerique peut-être considerée comme l'ensemble des pixels presentant une ou des propriétés données.

-REGLE DE DECISION:

La regle de decision définit la ou les propriétés nécessaires et suffisantes pour qu'un pixel (ou un ensemble de pixels) soit considéré comme appartenant à la classe correspondante.

-CLASSIFICATION MULTIDIMENSIONNELLE:

Dans les méthodes de classification multispectrale on agit au niveau des signatures spectrales et on réalise une classification dans un ensemble de points de R^n .

Une des possibilités pour ajouter la signature spatiale à la signature spectrale est de se ramener au problème précédent, en considérant la signature spatiale comme une ou des dimensions supplémentaires.

$$\begin{array}{l} X \left\{ \begin{array}{l} I_1(x,y) \\ I_2(x,y) \\ \vdots \\ I_n(x,y) \end{array} \right. \\ T \left\{ \begin{array}{l} T_1(x,y) \\ \vdots \\ T_m(x,y) \end{array} \right. \end{array}$$

2.ACQUISITION DE DONNEES:

L'obtention de ces informations nécessite:

- Des appareils de collecte de données,appelés CAPTEURS
- Des véhicules portant ces appareils,soient des plates-formes d'observation.
- Un système permettant la transmission des données au sol ou leur stockage à bord de la plate-forme.

a-LES CAPTEURS:

Le capteur est un dispositif qui reçoit les rayonnements électromagnétiques émis ou réfléchis par les objets au sol dans une certaine bande spectrale,les transforme en un signal électrique susceptible d'être amplifié,mesuré et stocké si nécessaire.

b-LES PLATES-FORMES D'OBSERVATION:

Suivant l'objectif visé,la collecte d'information doit s'effectuer à

différentes altitudes. On utilise pour cela trois types de plates-formes d'observation:

-LES SATELLITES

-LES BALLONS

-LES AVIONS

Leurs altitudes de vols sont respectivement:

500Km à 900Km pour les satellites de ressources terrestres, 10Km à 30Km pour les ballons et 500m à 10Km pour les avions.

Les satellites constituent donc la troisième plate-forme d'observation de la terre. Leur orbite régulière procure une stabilité avantageuse, par rapport aux avions et aux ballons, pour la collecte de données.

Leur altitude de vol élevée permet de couvrir un grand champ d'observation, néanmoins, ceci présente un inconvénient dû au fait que les radiations qu'ils captent sont, dans ce cas, affectée par l'ensemble des couches atmosphériques.

La résolution multispectrale des capteurs qu'ils transportent, permet de contourner ce problème.

La troisième partie de notre travail se basera sur des données prises à bord d'un satellite LANDSAT, par conséquent, nous nous intéresserons plus particulièrement à ce dernier.

3. LES SATELLITES LANDSAT:

3.1. HISTORIQUE:

Ils sont au nombre de trois, lancés respectivement en 1972, 1975 et 1978 à une altitude de 920Km. Ils suivent une orbite quasi circulaire. Ils effectuent 14 révolutions par jour d'une période de 103 minutes chacune. La première de chaque jour est décalée de 1,5 DEGRE vers l'ouest par rapport à celle de la veille. Ils assurent ainsi la couverture terrestre en 18 jours.

LE PREMIER (LANDSAT 1) a été lancé le 23 JANVIER 1972 par la NASA et a cessé son activité en JANVIER 1978.

LE DEUXIEME (LANDSAT 2) a été lancé le 22 JANVIER 1975 sur orbite de façon à produire un décalage de 9 jours sur la fréquence de passage des deux satellites (LANDSAT 1, LANDSAT 2) au dessus d'un même point du globe.

LE LANDSAT 2 permet donc d'étudier l'évolution d'un phénomène à des intervalles de temps réduits (9 jours au lieu de 18).

LE TROISIEME (LANDSAT3) a été lancé le 5 MARS 1978 dans le but de remplacer le LANDSAT 1 qui ne fonctionnait plus.

3.2.RESOLUTIONS SPECTRALES:

Les satellites LANDSAT sont équipés de deux types de capteurs: Les caméras R.B.V (Return-Beam-Vidicam) et les M.S.S (Multispectral scanner scanner).

a-Les caméras R.B.V sont au nombre de trois sur le LANDSAT 3.Elles photographient simultanément la même zone dans les trois bandes spectrales suivantes:

-Bande 4 : 0,46 - 0,60 micromètre.

-Bande 5 : 0,57 - 0,68 micromètre.

-Bande 6 : 0,66 - 0,82 micromètre.

b-Le M.S.S est un radiomètre multispectral à 4 canaux (4,5,6 et 7), munis de 24 détecteurs ,répartis au nombre de 6 par canal.Le M.S.S balaye une largeur de 185 km,à altitude normale (920km) en dessinant 6 lignes, correspondant aux 6 détecteurs,dans chacun des canaux simultanément.

b.1 RESOLUTION SPECTRALE DES M.S.S:

Les quatre types de détecteurs du système M.S.S travaillent respectivement dans les bandes spectrales suivantes:

-Bande 4: 0,50 - 0,60 micromètre.

-Bande 5: 0,60 - 0,70 micromètre.

-Bande 6: 0,70 - 0,80 micromètre.

-Bande 7: 0,80 - 1,10 micromètre.

b.2 CARACTERISTIQUES DES DIFFERENTES BANDES SPECTRALES:

-Bande 4: Située dans la partie visible du rayonnement électromagnétique elle permet l'identification des structures géologiques et la distinction entre les différents types de végétation.

-Bande 5: située dans le visible,elle est utilisée d'une part pour déterminer les caractéristiques topographiques et d'autre part pour classer les différents types de végétations.

-Bande 6: Située en partie dans le visible et en partie dans le proche infra rouge,elle

-Bande 7: Située entièrement dans l'infra-rouge, elle permet l'obtention de renseignements sur l'humidité (hydrologie) et sur les reliefs.

DEUXIEME

PARTIE

PROCESSUS DE RECONNAISSANCE DE FORMES

1. INTRODUCTION ET DEFINITIONS:

On définit comme forme toute structuration minimale perceptible qui émerge d'un bruit aléatoire, tout groupe d'éléments qui serait perçu, d'emblée avant que chaque détail doit isolé, comme n'étant pas le produit d'un assemblage au hasard.

La forme est aussi la manière dont les parties d'un tout sont arrangées entre elles, dont une chose est construite, disposée, agencée.

Le domaine de la reconnaissance des formes s'étend jusqu'à l'étude des structures mentales, logiques, d'une façon générale jusqu'à la recherche de toute trace d'organisation qu'elle soit dans un système quelconque.

Pour l'heure, les recherches s'occupent d'avantage du premier problème, plus simple, dont les motivations ont une urgence économique immédiate. En particulier, la robotique industrielle occupe dans ce domaine une place remarquable.

Comme méthodes, on peut classer celles-ci en deux approches générales:

- L'une statistique fondée sur la théorie statistique de décision.
- L'autre structurelle, fondée sur la théorie des langages et des automates.

L'approche statistique est la plus ancienne et la mieux établie.

Elle comporte trois étapes:

-LA PARAMETRISATION

-L'APPRENTISSAGE

-LA CLASSIFICATION

C'est ce qui a été déjà étudié précédemment

L'approche structurelle est celle qui est de nos jours l'objet de travaux.

2. ACQUISITION ET TRAITEMENT D'INFORMATION:

Nous nous limitons à des scènes dont les composants peuvent être assimilés à des formes bidimensionnelles.

Nous distinguons dans l'étape de traitement d'image trois phases conceptuelles: L'ACQUISITION, LA SEGMENTATION ET L'INTERPRETATION.

2.1. ACQUISITION DE L'IMAGE DE LA SCENE:

La phase d'acquisition transforme la scène visuelle en une image numérisée qui se présente sous forme d'un tableau bidimensionnel dont

chaque élément (pixel) porte l'information de luminosité d'une surface élémentaire de la scène originale, à la discrétisation des deux dimensions d'espace, s'ajoute une quantification de la luminosité dont les valeurs sont prises dans un ensemble fini de niveaux.

2.2. LA SEGMENTATION:

2.2.1. DETECTION DE CONTOURS:

On suppose que les objets présents dans la scène reposent sans recouvrement sur une surface plane.

Le problème est donc de reconnaître les formes des objets du contexte. La phase de segmentation consiste à partitionner l'image en un ensemble de régions à peu près uniforme et en un ensemble d'entités géométriques élémentaires, souvent des lignes.

2.2.2. RECHERCHE DU CONTOUR DES FORMES:

La détection de contour d'une région de l'image ou d'une forme, consiste à transformer l'image multi-niveaux en une image binaire, formée de 0 pour les points extérieurs à la forme et de 1 pour les points intérieurs. Le contour est défini par l'ensemble de points frontière entre forme et contexte.

3. METHODES:

3.1. METHODE DU GRADIENT:

L'opération gradient est une approximation de différentiation spatiale de l'échelle de gris. La méthode du gradient donne une image de contour ou image gradient. Les régions de l'image dont le niveau de gris est à peu près uniforme, sont caractérisées par un faible gradient, alors que les zones frontières de formes sont caractérisées par un gradient maximum.

3.2. METHODE DE HUECKEL:

Cette méthode procède de la façon suivante:

On déplace sur l'image une fenêtre circulaire de 40 à 60 points, en chaque position on pose les questions suivantes:

-Quelle est la droite qui sépare le mieux la fenêtre en deux plages uniforme?

-Est ce que la différence entre les niveaux de gris de ces deux plages ne dépasse pas un certain seuil?

La méthode permet donc la détermination du contour sous forme de segments

de droites, c'est une différence fondamentale avec la méthode du gradient qui, à partir d'une image (matrice de points) donne une image de contour.

3.3. METHODE A BASE STATISTIQUE:

Les méthodes à base statistique se basent sur l'analyse de la répartition des fréquences d'apparition des intensités lumineuses, obtenu par un HISTOGRAMME.

La forme dont on recherche le contour se caractérise par des niveaux de gris différents de ceux du contexte, l'histogramme fait donc apparaître deux modes ou pics caractérisant la forme et le contexte. La vallée située entre les deux pics permet donc de séparer la forme de son contexte. Ainsi le contour est obtenu en seuillant l'image par la valeur " T " du fond de la vallée.

3.4. UTILISATION DE LA METHODE DU GRADIENT:

La détermination des points frontières par l'exploitation de la technique du gradient présente un grand avantage en raison de la simplicité des algorithmes mis en oeuvre et par l'effet de l'opération gradient qui accentue les hautes fréquences spatiales riches en information sur les contours.

a-GRADIENT SIMPLE:

L'image gradient simple peut être calculée à partir de l'image initiale en appliquant l'opérateur de ROBERTS sur chaque fenêtre (2x2)

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Pour chaque point, le gradient est obtenu par l'équation

$$G(a) = |a-d| + |c-b|$$

Ainsi les zones frontières apparaîtront comme les zones gradient maximum et les zones homogènes comme les zones à gradient minimum, mais l'apparition d'un point bruité dans une fenêtre d'une zone homogène donne un gradient élevé, d'où l'inconvénient de cette méthode.

b-GRADIENT AMELIORE:

Dans cette méthode, l'image gradient est obtenue en calculant pour chaque fenêtre (4x4) de l'image.

Le gradient amélioré est donné par l'expression

$$G(a) = (A \cdot B \cdot C \cdot D)^{1/4} \quad \text{avec:}$$

$$\begin{array}{l}
 A = |f-k| + |j-g| \\
 B = |a-p| + |m-d| \\
 C = |b-o| + |i-h| \\
 D = |c-n| + |e-l|
 \end{array}
 \begin{array}{|c|c|c|c|}
 \hline
 a & b & c & d \\
 \hline
 e & f & g & h \\
 \hline
 i & j & k & l \\
 \hline
 m & n & o & p \\
 \hline
 \end{array}
 \text{fen\^etre } (4 \times 4)$$

Chaque point de la fen\^etre est utilis\^e une et une seule fois.

Le gradient am\^eliore consist\^e en un produit de 4 gradients conventionnels dans trois directions diff\^erentes.

L'image gradient n'est pas sensible \^a la rotation de l'image.

Les points du contour seront mis en \^evidence, car le produit de gradient de grandes valeurs est tr\^es grand. Au contraire l'ensemble des points isol\^es d\^u aux bruits et \^a la differentiation est consid\^erablement r\^eduit car l'un au moins des 4 gradients A, B, C, D a une valeur faible, le produit est donc \^egalement faible.

c-TRACE D'HISTOGRAMME DE GRADIENT:

Ceci montre qu'un plus grand nombre de points le long des contours apparait au pic d'histogramme. (fig 1)

4. AFFINAGE ET LISSAGE DES LIGNES DE CONTOUR:

Les contours qui resultent de l'application de la m\^ethode du gradient am\^eliore sont bruit\^es et ont par endroit une certaine \^epaisseur. Cette \^epaisseur est d\^ue \^a la multiplication des points voisins du contours dans les zones \^a transitions d\^egrad\^ees.

D'o\^u une importante phase de filtrage et de lissage des contours qui ralentit l'algorithme pr\^ec\^edent.

4.1. FILTRAGE ANALYTIQUE:

Un premier filtrage analytique permet par seuillage de r\^eduire le bruit dans l'image gradient am\^eliore.

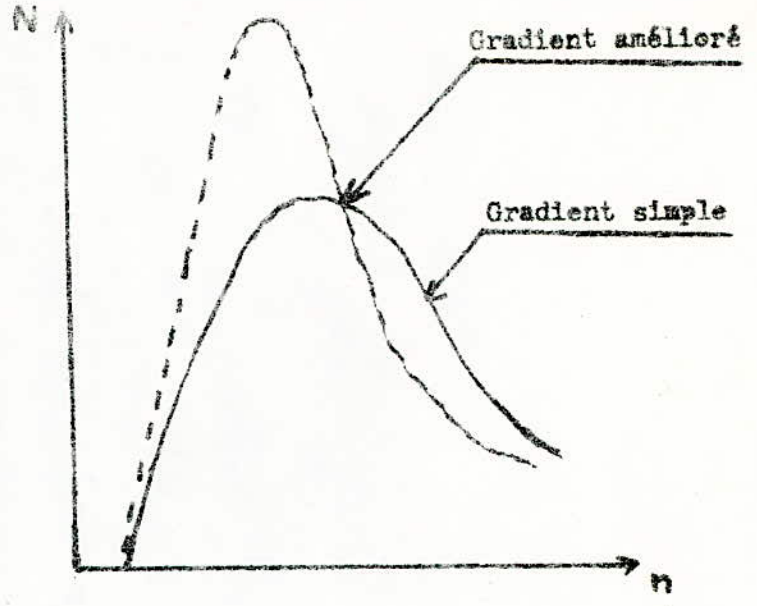
Apr\^es avoir calcul\^e une moyenne locale M sur une fen\^etre $(2L+1 \times 2L+1)$ centr\^e au point courant (x,y) , on proc\^ede \^a l'\^elimination de ce point si la valeur de son gradient am\^eliore est inf\^erieur \^a la moyenne de la fen\^etre.

$$g^*(x,y) = \begin{cases} 0 & g(x,y) < M \\ 1 & \text{si non} \end{cases}$$

Avec:

$$M = \frac{1}{(2L+1)^2} \sum_{u=-L}^{u=L} \sum_{v=-L}^{v=L} g(x+u, y+v)$$

fig.1



n: valeur du gradient

N: fréquence de cette valeur

* $g(x,y)$: point de l'image gradient ~~calculé~~

$g(x,y)$: point de l'image originale

Ce filtrage transforme l'image en image binaire.

4.2. LISSAGE DES LIGNES DE CONTOUR:

Cette opération permet d'affiner les lignes de contour, en traitant les fenêtres binaires (3x3) centrées aux points dont la valeur est un(1).

Si ces fenêtres satisfait à l'une des configuration suivantes, la valeur du centre devient zéro(0).

1 X 0	X 1 1	X 1 X	X 1 X
1 (1) 0	0 (1) X	1 (1) X	X (1) 1
X 0 0	0 0 0	X X 0	0 X X
(1)	(2)	(3)	(4)

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

REMARQUE:

X: peut prendre la valeur zéro(0) où un(1).

EXEMPLES:

1)-	1000000	1000000
	0110000	0110000
	0011000	0001000
	0000111	0000111
	Avant lissage	Après lissage
2)-	100	100
	100	100
	110	010
	110	010
	110	010
	010	010
	010	010
	Avant lissage	Après lissage

5. TRAITEMENT D'UNE IMAGE PHOTOGRAPHIQUE AERIENNE DIGITALISEE:

L'objet de ce paragraphe est le traitement numérique d'un échantillon de photographie aérienne. Sur cet échantillon nous établirons les algorithmes de reconnaissance de formes et de détection de contours.

Les différentes étapes de notre processus sont:

- Obtention de l'image numérique
- Enregistrement des données de l'image (niveau de gris)
- Transformation de cette image en une image gradient
 - .Gradient simple
 - .Gradient amélioré
- Filtrage de l'image gradient
- Affinage et lissage
- Représentation graphique

5.1. OBTENTION DE L'IMAGE NUMERIQUE:

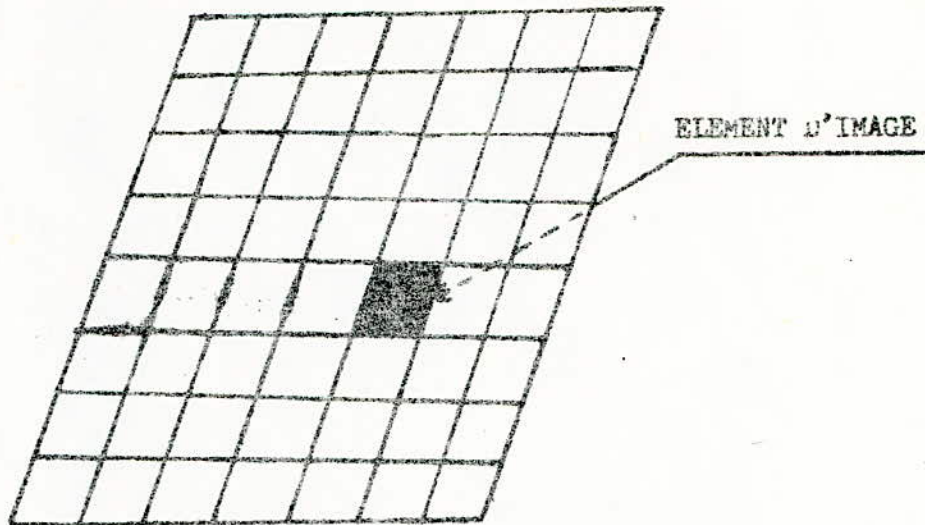
La numérisation d'une image photographique se fait point par point. Pour cela l'image est divisée en surfaces élémentaires dont les dimensions dépendront de la précision de l'appareil de numérisation utilisé. (fig.2)

Dans notre cas l'image a été déjà digitalisée et elle est de dimension (30x60) soit 1800 point. Chaque point représentant une valeur de ton de gris.

5.2. ENREGISTREMENT DES DONNEES:

Les valeurs numériques obtenues par digitalisation constituent les

fig.2



données de l'image, elles représentent les informations nécessaires à son traitement. Ces données se présentent sous forme d'une matrice. Afin de pouvoir les manipuler par ordinateur, il est nécessaire de les enregistrer sur une de ses unités périphériques (bande magnétique, disque ou cartes perforées). Dans le cas de l'échantillon choisi ici, le nombre d'information à traiter étant relativement réduit, nous avons opté pour l'enregistrement sur cartes perforées.

5.3. TRANSFORMATION DE L'IMAGE EN UNE IMAGE GRADIENT:

a-GRADIENT SIMPLE:

Pour obtenir l'image gradient, on établit le programme qui consiste à:

- IMPRIMER LA MATRICE DE DONNEES
- CALCULER ET IMPRIMER L'IMAGE GRADIENT
- CODER L'IMAGE:

Pour cela on choisit un seuil et on compare la valeur du gradient de tous les points à ce dernier. Si le gradient est supérieur au seuil, on le remplace par un point (.), si non par un blanc ().

b-GRADIENT AMELIORE:

On procède de la même manière, et on compare le résultat obtenu avec le précédent.

c-COMPARAISON:

D'après les résultats obtenus, nous constatons que la méthode du gradient amélioré, met plus en évidence les contours, ainsi qu'elle réduit l'effet des points bruités.

5.4. CONCEPTION DU PROGRAMME DE DETECTION DE CONTOURS:

Ce programme est conçu pour détecter et tracer les contours d'une image numérique dont les valeurs sont enregistrées sur cartes perforées.

STRUCTURE DU PROGRAMME: (fig. 3)

-Le programme principal permet de réserver l'espace mémoire nécessaire au travail à effectuer, et d'appeler les différents sous-programmes.

-Le sous-programme GRADA permet de calculer le gradient amélioré de l'image.

-Le sous-programme IEIN permet de filtrer l'image transformée.

-Le sous-programme TRANS son rôle est de résoudre le problème de traitement aux limites, pour cela on inclut l'image binaire (30x60) dans une image (32x62), dont ses éléments sont des zéros (0).

-Le sous-programme ILISS permet d'affiner et de lisser les contours.

-Le sous-programme ICOD permet de tracer les contours.

REMARQUE:

Toutes les fonctions d'entrées-sorties sont indiquées sur le tableau de la figure. 4.

Fig.3

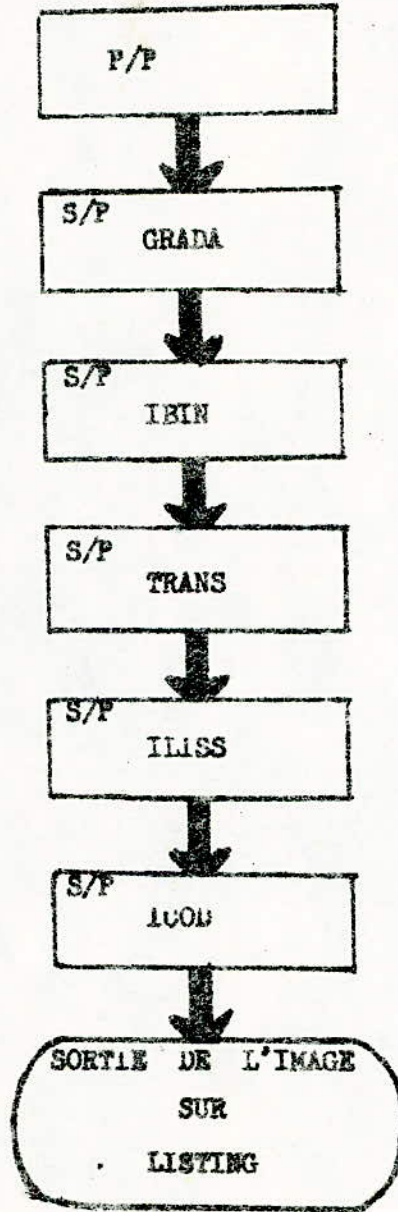


fig.4

NOMS DES SUB-ROUTINES	FONCTIONS	ENTREES	SORTIES
GRABA	calcul du gradient de chaque point de l'image	matrice des données de l'image Nombre de ligne Nombre de colonne	matrice gradient (N,L)
IRIN	filtrage de l'image gradient fenetre choisie (5x5)	matrice gradient (N,L)	matrice binaire (N,L)
TRANS	resolution du problème aux limites	matrice binaire (N,L)	matrice binaire (N+2,L+2)
LISS	affinage et lissage des lignes de contours	matrice binaire (N+2,L+2)	matrice binaire lissée (N+2,L+2)
ICON	tracé les lignes de contours	matrice binaire lissée (N+2,L+2)	image codée par des points et des blancs

MATRICE DE DONNEES

Matrix of data with 27 columns and 27 rows. The matrix contains a complex pattern of numbers and characters, including digits 0-9, letters A-Z, and symbols like \$ and %. The pattern is dense and appears to be a form of data encoding or a specific matrix representation.

IMAGE GRADIENT SIMPLE

Matrix of data for image gradient, with 27 columns and 27 rows. The matrix contains a complex pattern of numbers and characters, including digits 0-9, letters A-Z, and symbols like \$ and %. The pattern is dense and appears to be a form of data encoding or a specific matrix representation.

TROISIEME

PARTIE

I) TEXTURE DES IMAGES DE TELEDETECTION.

La signature spectrale d'un point d'une image est habituellement considérée comme l'information caractéristique de chaque pixel. Or, caractériser complètement un pixel c'est ajouter à la démarche précédente, la description de l'environnement de ce pixel c'est à dire l'information spatiale dans son voisinage. Cette dernière peut se diviser en deux groupes : la structure, qui décrit la manière dont les objets de l'image sont arrangés entre eux, et, la texture, qui exprime la distribution des tons de gris à l'intérieur de ces objets.

1) Définition de la texture :

Nous appelons " information de texture d'un pixel " l'information apportée par l'ensemble des pixels appartenant à un voisinage approprié de celui-ci.

Si nous considérons maintenant une région d'image définie par ses frontières, nous appellerons " texture d'une région donnée " les caractéristique de répartition spatiale des niveaux de gris des pixels dans cette région, d'où la notion de spatialité et de localité.

L'idée de spatialité implique que la texture n'est pas liée au ton de gris d'un point mais aux relations spatiales entre points voisins ou entre objets élémentaires de petite taille. Ces objets élémentaires ne sont pas considérés comme des individualités isolées mais seulement comme composantes d'un motif.

L'idée de localité implique que les objets élémentaires composant la texture doivent être de petite taille et décrit le comportement de la distribution des tons de gris sur le voisinage de chacun des points de l'image.

Une approche plus globale consiste, dans les définitions suivantes, à lier la définition de la texture au processus d'interprétation. Le photo-interprète définit un ensemble de texture sur l'image comme le mathématicien définit une structure sur un ensemble

Trois postulats intuitifs sont utilisés (Guy 1970)

- Il est possible d'isoler des " formes " en les distinguant par un moyen quelconque.

- Chacun de ces formes se répètent, est reconnaissable et permet ainsi la définition des classes d'éléments.

- Ces classes d'éléments peuvent être organisées entre elles ou entre sous classes.

Un élément textural est une de ces formes particulières reconnue comme faisant partie d'une classe qui est une texture, la structure est alors une propriété caractéristique de l'organisation des classes (Postulat 3).

Mathématiquement, le premier postulat définit, à partir de l'image, un ensemble \mathbb{E} (Braconne 1975), le second postulat définit une relation d'équivalence \mathcal{R} sur l'ensemble \mathbb{E} de " formes " dont les classes d'équivalence ont plus d'un élément, le troisième postulat exprime simplement l'existence de l'ensemble des parties de \mathbb{E} .

Au total, une texture devient une classe d'équivalence (élément de l'ensemble quotient \mathbb{E}/\mathcal{R}). Un élément textural, le couple $(e, \hat{e}) \in \mathbb{E} \times \mathbb{E}/\mathcal{R}$, et, une structure est un sous ensemble de $\mathcal{P}(\mathbb{E} \times \mathbb{E}/\mathcal{R})$.

L'ensemble \mathbb{E} est donc composé, dans notre définition, de parties de D (domaine spatial de l'image) de petite taille formant un recouvrement de D , chaque partie P de D , élément de \mathbb{E} , est affectée d'un coefficient vectoriel $m(P) = (m_1, \dots, m_k) \in \mathbb{N}^k$ qui mesure la distribution spatiale des tons de gris sur P . Les relations d'équivalence définies sur \mathbb{E} ne tiennent compte que des propriétés spatiales locales exprimées par m . Ce sont, en général, des relations déduites de la partition de \mathbb{E} par m^{-1} .

$$P_1 \mathcal{R} P_2 \iff m(P_1) = m(P_2)$$

$$\iff \exists n \in \mathbb{N}^k \text{ tq } P_1 \text{ et } P_2 \in m^{-1}(n)$$

Une texture est alors une classe de parties de D ayant localement des caractéristiques semblables de distribution de tons gris. Une valeur particulière de m caractérise une texture.

2) PLACE DE LA TEXTURE DANS L'INTERPRETATION DE DIFFERENTES IMAGES DE TELEDETECTION:

La texture est un élément d'analyse utilisé dans plusieurs domaines:

- Domaine médical: pour l'étude automatique de radiographies (SUTTON et HALL, 1972), pour l'étude de photographies microscopiques de chromosomes (ROSENFELD, 1972).
- Domaine métallurgique: Pour l'étude microscopique des impuretés des métaux.
- Domaine de teledetection: Pour l'étude d'images de la surface terrestre. Dans notre cas, nous étudierons des images prises à partir de satellites LANDSAT.

La texture des images satellites permet de distinguer de grandes unités de paysage: l'arrangement de l'espace agricole, par exemple.

HARRALICK (1973) a pu appliquer avec succès des méthodes d'analyse texturale à des images de satellite, reconnaissant des forêts cotières, des bois, des prairies, des villes et des zones urbaines.

3) TECHNIQUES NUMERIQUES D'ANALYSE DE LA TEXTURE:

Les techniques d'analyse de la texture d'images sont très nombreuses utilisant des méthodes optiques ou numériques. Nous présenterons les techniques numériques les plus importantes en distinguant celles dont le but est de quantifier la notion de texture.

3.1. ETUDE DE QUELQUES SIGNATURES DE TEXTURE:

Dans le but de classer les différentes méthodes d'analyse de la texture, HARRALICK et AL distinguent des méthodes classées en plusieurs groupes.

Le premier groupe comprend trois méthodes plus ou moins liées à la mesure des fréquences spatiales d'une sous-image: fonction d'autocorrelation, transformées orthogonales et mesure de densité de frontières.

a-La méthode utilisant les transformations orthogonales telles que les transformations de Fourier, d'Hadamard...

La sous-image à classer, d'abord représentée par les tons de gris de ses points, et réexprimée sur de nouveaux axes reliés à la fréquence spatiale.

b-Dans la méthode utilisant la mesure de la densité de frontières, les hautes fréquences sont traduites par une densité de frontières élevée indiquant la présence d'une texture fine.

D'autres groupes mettent en évidence la méthode des éléments structuraux où est détectée, sur une image binaire la disposition régulière de formes élémentaires (éléments structuraux).

La dernière méthode est celle des matrices de co-occurrence où l'on

mesure, pour chaque sous-image, la probabilité de rencontre de deux tons de gris dans une relation spatiale donnée.

Deux méthodes ont été appliquées à la texture et elles sont bien différentes.

Les matrices de co-occurrence et les méthodes utilisant les transformations locales.

3.1.1. LES MATRICES DE CO-OCCURENCE:

L'idée maîtresse de cette méthode est de supposer que "l'information texturale d'une image est contenue dans les relations spatiales globales ou moyennes que les tons de gris de l'image ont entre eux"

On construit pour la sous-image étudiée, une matrice $P = (p_{ij})$ où p_{ij} est le nombre de fois que les tons de gris i et j entrent dans une relation spatiale donnée ("voisin de" dans une direction donnée).

Plus précisément, P_d est la matrice de probabilité de rencontre des tons de gris i et j pour des points distants de d et formant un angle (θ) avec l'horizontale.

Considérons un exemple d'image (4×4) de ton de gris entre 0 et 3.

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

La matrice de co-occurrence $P_d(i, j)$ dans la direction horizontale ($d = (1, 0)$) est:

I	0	1	2	3
0	4	2	1	0
1	2	4	0	0
2	1	0	6	1
3	0	0	1	2

C'est de cette matrice que l'on extrait les paramètres texturaux, pour cela 14 indices de texture sont calculés pour chacune des 4 directions, ou bien moyennés sur ces directions.

Nous avons les indices suivants:

$$f_1 = \sum_{i=1}^N \sum_{j=1}^N P_{ij}^2$$

$$f_2 = \sum_{n=0}^{N-1} n^2 \left(\sum_{i-j=n} P_{ij} \right)$$

$$f_3 = \frac{\sum_i \sum_j (ixjxP_{ij} - \mu_x \mu_y)}{\sigma_x \sigma_y} \quad (\text{corrélation})$$

$$f_4 = \sum_i \sum_j (i-j)^2 x P_{ij} \quad (\text{variance})$$

$$f_5 = \sum_i \sum_j \frac{1}{1 + (i-j)^2} P_{ij} \quad (\text{moment des différences inverses})$$

$$f_6 = \sum_{k=2}^{2N_G-1} k \cdot P_{x+y}(k) \quad \text{avec} \quad P_{x+y}(k) = \sum_i \sum_j P(i,j) \quad i+j=k$$

(moyenne des sommes)

$$f_7 = \sum_{k=2}^{2N_G-1} (k - f_6)^2 P_{x+y}(k) \quad (\text{variance des sommes})$$

$$f_8 = - \sum_{k=2}^{2N_G-1} P_{x+y}(k) \log (P_{x+y}(k)) \quad (\text{entropie de la somme})$$

$$f_9 = - \sum_i \sum_j P(i,j) \log (P(i,j)) \quad (\text{entropie})$$

$$f_{10} = \sum_{k=0}^{N_G-1} (k - \mu_{x-y})^2 P_{x-y}(k) \quad \text{avec} \quad \mu_{x-y} = \frac{1}{N_G} \sum_{k=0}^{N_G-1} P_{x-y}(k)$$

(variance des différences) et $P_{x+y} = \sum_i \sum_j P(i,j) \quad \text{avec} \quad i+j=k$

$$f_{11} = - \sum_{i=0}^{N_G-1} P_{x-y}(i) \log (P_{x-y}(i)) \quad (\text{entropie des difference})$$

$$f_{12} = \frac{HXY - HXY1}{\max(HX, HY)} \quad (\text{deux mesures de corrélation de l'information})$$

$$f_{13} = (1 - \exp(-2.0(HXY2 - HXY)))^{1/2} \quad (f_{12} \text{ et } f_{13})$$

avec $HXY = f_9$

$$HX = - \sum_{k=0}^{N_G-1} P_x(k) \log(P_x(k))$$

$$HY = - \sum_{k=0}^{N_G-1} P_y(k) \log(P_y(k))$$

$$HXY1 = - \sum_i \sum_j P(i,j) \log(P_x(i)P_y(j))$$

$$HXY2 = - \sum_i \sum_j P_x(i)P_y(j) \log(P_x(i)P_y(j))$$

$$\text{et } P_x(i) = \sum_{j=1}^{N_G} P(i,j)$$

$$P_y(j) = \sum_{i=1}^{N_G} P(i,j)$$

$$f_{14} = (\text{deuxième valeur propre la grande de la matrice } Q(i,j).)^{1/2}$$

$$\text{avec } Q(i,j) = \sum_{k=1}^{N_G} \frac{P(i,j) \cdot P(k,j)}{P_x(i) \cdot P_y(j)}$$

(coefficient de corrélation maximum)

N_g : est le nombre de tons de gris

P_{ij} : matrice de co-occurrence normalisée

les moyennes et écarts types des distributions marginales associées à P

f_1 : moment angulaire du second ordre croissant avec l'homogénéité de l'image

f_2 : mesure du contraste

f_3 : mesure de corrélation

f_9 : entropie mesurant le degré de désordre de l'image

La richesse de cette méthode consiste dans le nombre d'aspects de la texture reflétés par ces 14 indices, qui ne sont pas obligatoirement utilisés ensemble mais parmi lesquels, on peut effectuer un choix suivant le type d'application envisagé.

Plusieurs de ces indices peuvent se recouper, ainsi f_9 (entropie) appliqué à plusieurs sous-images tests donne un ordre de classement inverse du moment angulaire (f_1).

1.3.2. TRANSFORMATIONS LOCALES, DENSITÉ DE FRONTIÈRES :

Dans la méthode utilisant les matrices de co-occurrence, on étudie les relations spatiales entre tons de gris (espace spectral). À l'inverse les méthodes utilisant les transformations locales utilisent les relations spectrales entre points (espace géométrique).

Ces méthodes comparent les tons de gris de deux points, les tons de gris étant ceux de l'image originale ou ceux résultant d'un traitement antérieur.

Precisons dans ce cas que:

- La comparaison consiste simplement à faire la différence entre les tons de gris (opérateur local: gradient).
- Les deux points comparés seront dans une relation spatiale donnée (par exemple: voisins horizontaux).
- Le ton de gris du point reflétera alors une propriété locale du voisinage du point sur l'image originale.

-ROSENFELD (1969) a proposé un algorithme simple qui consiste à remplacer la valeur de point par la différence de deux points de son voisinage sur deux cotés opposés.

Soit $I(x-1,y)-I(x+1,y)$ pour la direction horizontale. Cette mesure du gradient est ensuite moyennée sur le voisinage de (x,y) , ce qui donne un paramètre textural pour le point (x,y) .

-SUTTON et HALL (1972) étendent cette notion de gradient local, ils calculent pour chaque sous-image étudiée:

$$g(d) = \sum_{i,j} \left[|I(i,j)-I(i+d,j)| + |I(i,j)-I(i-d,j)| \right. \\ \left. + |I(i,j)-I(i,j+d)| + |I(i,j)-I(i,j-d)| \right]$$

L'étude du graphe de $g(d)$ révèle les propriétés texturales de la sous-image.

-ROSENFELD et AL (71,72) proposent des algorithmes fondés sur des combinaisons de gradients de plusieurs tailles et sur des images moyennées pour détecter des frontières, des taches ou des traits d'une image.

DETECTION DE FRONTIERES:

La detection de frontières se fait en 4 étapes:

1) Calcul de moyennes sur voisinages variables de taille $2^k \times 2^k$

$$A_k(x,y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} a(i,j) / (2^k \times 2^k)$$

2) Pour chaque taille k , pour chaque point (x,y) , calcul des différences entre voisinages de même direction de chaque côté du point (x,y) . Ceci est calculé pour les quatre directions : horizontale, verticale, première et deuxième diagonales.

Par exemple, pour la direction horizontale:

$$E_{kH}(x,y) = A_k(x+2^{k-1}, y) - A_k(x-2^{k-1}, y)$$

On prend alors la valeur maximale de ces quatre différences, ce qui détermine la meilleure orientation.

3) Pour chaque point, choix de la meilleur taille k qui est la plus

grande taille telle que la taille inferieure $k-1$ donne:

$$E_k \gg E_{k-1} \text{ de } 75\%$$

4) Elimination de tout point qui a dans son voisinage et dans une direction perpendiculaire à sa meilleure orientation un gradient plus élevé.

DETECTION DE TACHES:

Remplacer l'étape 2 par l'étape suivante:

Pour chaque point (x,y) et chaque taille de voisinage, on effectue les differences entre $A_k(x,y)$ et les voisinages de même taille autour du point. Les valeurs positives de ces differences detectent la presence d'une tache.

DETECTION DE TRAITS:

Pour chaque taille de voisinage, on effectue les differences entre $A_k(x,y)$ et les moyennes sur des paires de voisinages de même taille et de côtés opposés de (x,y) , ceci dans toutes les directions. Si une paire de differences, pour l'orientation θ , est positive et si ceci est vrai aussi en remplaçant (x,y) par les deux points de son voisinage situés dans la direction $\theta + \frac{\pi}{2}$, on prend la moyenne de ces 3 paires de differences comme valeur du detecteur de traits.

Les deux méthodes (matrices de co-occurrence et transformation locale) sont d'une richesse d'interprétation égale, les matrices de co-occurrence par le nombre d'indices f_1 exprimant divers aspects de la notion de texture et ces transformations locales, par les multiples possibilités de combinaisons de gradient dans diverses directions et tailles.

Les matrices de co-occurrence possèdent toute fois une plus grande lourdeur due aux calculs d'indices sur des blocs assez importants entrainant donc une perte de resolution (temps machine).

De plus, les méthodes utilisant les transformations locales présentent, pour nous, une plus grande facilité d'implantation qui se traduit aussi par un coût moins important.

C'est donc cette dernière approche que nous avons développé en montrant les diverses possibilités d'utilisation des transformations locales de plus simples aux plus complexes.

2. TRANSFORMATIONS LOCALES:

2.1. INTERET DES TRANSFORMATIONS LOCALES POUR L'ETUDE DE LA TEXTURE:

Ce chapitre sera consacré à la définition et à l'étude d'une formulation mathématique de la texture. Cette formulation utilise un ensemble d'applications qui modifient le ton de gris d'un point en fonction des tons de gris de ses voisins d'où leur nom de " TRANSFORMATIONS LOCALES ".

La manière dont on tient compte du voisinage met en évidence sur l'image dérivée différentes caractéristiques de la répartition locale des tons de gris.

Ces transformations locales sont adaptées à la quantification de texture: Elles expriment en effet les rapports du ton de gris du point avec ceux de ses voisins, ce qui résume les idées de spatialité et de localité comprises dans la définition de la texture.

D'autre part, les transformations locales sont nombreuses, offrant une large gamme de possibilités de quantification, en général très simples, donc d'une implantation facile et d'un coût informatique peu élevé.

Notons qu'il y a trois possibilités d'utilisation des transformations locales:

Une transformation T_n s'écrit:

$$T_n: I(x,y) \longrightarrow I_n(x,y)$$

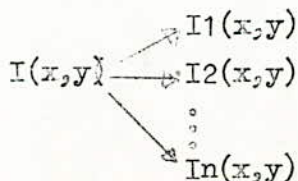
I: Image originale

. I_n : Image transformée

L'étude de la texture utilise les transformations locales de trois manières principales:

a. Etude directe de l'image transformée I_n visualisée sur imprimante, console graphique ou écran, cela en vue d'une aide à l'interprétation visuelle ou d'un traitement automatique ultérieur.

b. Classification multidimensionnelle où chaque dimension est obtenue en appliquant à l'image originale I une transformation T_n .



A chaque point correspond un vecteur à n dimensions $(I(x,y) \dots I_n(x,y))$ qui représente la texture de son voisinage.

c. Classification multidimensionnelle où chaque dimension est un paramètre textural t_i issu de la comparaison entre les n transformées locales.

$$\begin{aligned} (x,y) &\longrightarrow (I_1(x,y), I_2(x,y), \dots, I_n(x,y)) \\ &\longrightarrow (t_1(x,y), t_2(x,y), \dots, t_k(x,y)) \end{aligned}$$

avec par exemple:

$$t_i(x,y) = \max(I_1(x,y), \dots, I_n(x,y))$$

Dans chaque cas, l'ensemble E, la relation R et la mesure m revêtent une forme différente.

E est l'ensemble des voisinages des points de l'image I_n , m est le scalaire $I_n(x,y)$.

$$\begin{aligned} E &= \{ (V(x,y), I_n(x,y)) \text{ tq } (x,y) \in D \} \\ &= \\ &= \{ e(x,y) \text{ tq } (x,y) \in D \} \end{aligned}$$

2.2. TRANSFORMATIONS LOCALES: DEFINITIONS-THEOREME.

2.2.1. IMAGE DIGITALE NUMERISEE:

Définition: C'est une application de $D = I_C \times I_L \subset \mathbb{N} \times \mathbb{N}$ dans $T \subset \mathbb{N}$.
 T, I_C, I_L étant des sous ensembles de \mathbb{N} de la forme $\{1, 2, \dots, n\}$

$$I: (x,y) \in D \longrightarrow t = I(x,y) \in T$$

-D est le domaine spatial de l'image.

- I_C et I_L déterminent respectivement le nombre de colonnes et de lignes.

(x,y) est le point de résolution élémentaire de l'image..

-t est le ton de gris du point.

2.2.2. VOISINAGE:

Définition: $V(x,y)$ est un voisinage du point (x,y) si et seulement si:

$$\bullet V \in P(D)$$

$$\bullet (x,y) \in V(x,y) \quad \forall (x,y) \in D$$

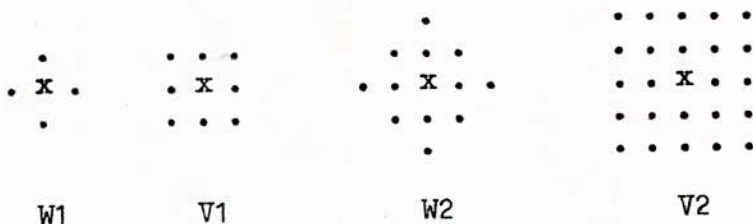
$$\bullet (x',y') \in V(x,y) \Rightarrow (x,y) \in V(x',y')$$

Par la suite, les voisinages utilisés sont les sous ensembles de $P(D)$

$\{V_k\}$ et $\{W_k\}$ définis par les expressions suivantes:

$$V_k(x,y) = \{x-k, x-k+1, \dots, x+k\} \times \{y-k, y-k+1, \dots, y+k\}$$

$$W_k(x,y) = \{x-k, x+k\} \times \{y-k, y+k\} + V_{k-1}(x,y)$$



L'ensemble des parties ainsi défini permet de construire des concepts analogues à ceux de la Topologie générale:

Ouvert, fermé, intérieur . . .

-INTERIEUR D'UN SOUS ENSEMBLE A DE D :

Soit $A \in (D)$, par définition, l'intérieur de A est le sous ensemble

$$\dot{A} = \{ a \in A \text{ tq } V(a) \subset A \}$$

-ENSEMBLE OUVERT:

$$A \text{ ouvert} \iff \dot{A} = A$$

-ENSEMBLE FERME:

$$A \text{ fermé} \iff \overset{\circ}{A} \text{ ouvert}$$

2.2.3. TRANSFORMATION LOCALE:

a) Définition.

Soit un voisinage V défini sur le domaine D d'une image I par

$$(x_0, y_0) \in D \implies V(x_0, y_0) = \{ (x_0, y_0), (x_1, y_1), \dots, (x_n, y_n) \}$$

Une transformation locale sur le voisinage V est une application

de l'ensemble I des images sur D dans lui-même défini par

$$I \implies \tilde{I} = I'$$

$I': (x, y) \in D \implies K(I(x_0, y_0), I(x_1, y_1), \dots, I(x_n, y_n))$ où K est un opérateur de N dans N .

I: image originale

I': image transformée de I

Une transformation locale agit uniformément sur toute l'image, d'où son intérêt pour le traitement et la reconnaissance des zones texturales.

Cela sera précisé et démontré dans le paragraphe qui suit. Une classe particulière de transformations locales sera étudiée: Les T.L linéaires.

b) Invariance en position:

Définition (ROSENFELD 69):

Une transformation ζ de \mathbb{C} dans lui-même est dite invariante en position si, T_{kl} étant la transformation définie sur \mathbb{C} par:

$$T_{kl}(I)(x,y) = I(x-k, y-l)$$

On a :

$$\forall I \in \mathbb{C} \quad (T_{kl}(I)) = T_{kl}(\zeta(I))$$

Une transformation invariante en position agit uniformément sur toute l'image.

THEOREME:

Toute transformation locale est invariante en position

$$\begin{aligned} \zeta \left[T_{kl}(I) \right] &= \underbrace{K(T_{kl}(I)(x_0, y_0), T_{kl}(I)(x_1, y_1), \dots, T_{kl}(I)(x_n, y_n))}_{\text{sur } V(x_0, y_0)} \\ &= \underbrace{K(I(x_0-k, y_0-l), I(x_1-k, y_1-l), \dots, I(x_n-k, y_n-l))}_{\text{sur } V(x_0-k, y_0-l)} \\ &= T_{kl}(\zeta(I)) \end{aligned}$$

c) Transformations locales linéaires:

(i) Définition:

Une transformation ζ de \mathbb{C} dans lui-même est linéaire si et seulement si:

$$\begin{aligned} \forall I_1, I_2 \in \mathbb{C} \\ \forall a, b \in \mathbb{N} \quad \zeta(aI_1 + bI_2) = a\zeta(I_1) + b\zeta(I_2) \end{aligned}$$

(ii) Expression d'une transformation locale linéaire:

Dans ce paragraphe, la T.L. est définie sur un voisinage V_n , ceci pouvant être facilement généralisé.

THEOREME:

Soit ζ une T.L. linéaire définie sur le voisinage V_n et I' l'image transformée de I par ζ , alors, il existe des coefficients entiers

$((a_{k', l'}, k' = -n, \dots, +n), l' = -n, \dots, +n)$ tels que:

$$I'(x,y) = \sum_{k'=-n}^{+n} \sum_{l'=-n}^{+n} a_{k', l'} I(x-k', y-l')$$

Démonstration:

Toute image peut s'écrire sous la forme

$$I = \sum_{k=1}^K \sum_{l=1}^L I(k,l) \cdot T_{kl}(I_0)$$

Où I_0 est la fonction de Z^2 dans N définie par:

$$I_0(0,0) = 1$$

$$I_0(x,y) = 0 \quad \text{si } (x,y) \neq (0,0)$$

Soit une transformation locale linéaire :

$$\begin{aligned} \tilde{\mathcal{E}}(I) &= \tilde{\mathcal{E}} \left(\sum_k \sum_l I(k,l) \cdot T_{kl}(I_0) \right) \\ &= \sum_k \sum_l I(k,l) \tilde{\mathcal{E}}(T_{kl}(I_0)) \end{aligned}$$

puisque $\tilde{\mathcal{E}}$ est invariante en position

$$\tilde{\mathcal{E}}(I) = \sum_k \sum_l I(k,l) T_{kl}(\tilde{\mathcal{E}}(I_0))$$

posons $I'_0 = \tilde{\mathcal{E}}(I_0)$

$$\text{alors } I'(x,y) = \sum_k \sum_l I(k,l) I'_0(x-k, y-l) \quad (1)$$

faisons le changement de variable

$$\begin{cases} k' = x-k \\ l' = y-l \end{cases} \quad I'(x,y) = \sum_{k'} \sum_{l'} I(x-k', y-l') \cdot I'_0(k', l') \quad (2)$$

l'expression garde un sens en posant $I(x,y) = 0$ si (x,y) est en dehors du domaine de l'image.

posons

$$a_{k',l'} = I'_0(k', l'), \quad I' \text{ s'écrit donc}$$

$$I'(x,y) = \sum_{k'} \sum_{l'} I(x-k', y-l') a_{k',l'}$$

ce qu'il fallait démontrer.

Plus précisément:

$$\begin{aligned} I'(k', l') &= \tilde{\mathcal{E}}(I_0)(k', l') \\ &= K(I_0(k', l'); I_0(k'-n, l'-n), \dots, I_0(k'-i, l'-j), \dots, \\ &\quad I_0(k'+n, l'+n)) \quad \text{sur } V_n(k', l') \end{aligned}$$

$$i = -n, \dots, +n$$

$$j = -n, \dots, +n$$

Le terme $I_0(k'-i, l'-j)$ est nul sauf pour

$$\begin{cases} i = k' \\ j = l' \end{cases}$$

On a donc deux cas:

$$\begin{aligned} 1^{\text{er}} \text{ cas:} \quad & -n \leq k' \leq +n \\ & -n \leq l' \leq +n \end{aligned}$$

$$I_0'(k', l') = K(0, 0, \dots, 1, 0, \dots, 0) \\ (i, j) = (k', l')$$

$$\begin{aligned} 2^{\text{ème}} \text{ cas:} \quad & k' > n \quad \text{ou} \quad k' < -n \\ & l' > n \quad \quad \quad l' < -n \end{aligned}$$

$$I_0'(k', l') = K(0, 0, \dots, 0) = 0 \quad \text{par linéarité}$$

Donc

$$a_{k', l'} = 0 \text{ si } (k', l') \in (-n, +n) \times (-n, +n)$$

$$a_{k', l'} = K(0, 0, \dots, 1, 0, \dots, 0) \text{ si } (k', l') \in (-n, +n) \times (-n, +n)$$

REMARQUE:

Les équations (1) et (2) appartiennent à un produit de convolution

$$\mathcal{F}(I) = I' = I' \otimes I_0'$$

La fonction I_0' et par conséquent la matrice $(a_{k', l'})$ déterminent la transformation locale \mathcal{F} .

La fonction I_0' , donne la réponse de la T.L. à un signal unité localisé en un point de l'image.

d) Domaine de l'image transformée:

L'expression de la transformée $I'(x_0, y_0)$ nécessite la connaissance de I sur le voisinage V du point (x_0, y_0) . Pour les points du bord de l'image la transformation n'est définie que sur le voisinage TRONQUÉ $V \cap D$.

Deux solutions sont possibles

-Les bords de l'image sont négligés en ne traitant que les points (x_0, y_0) tels que $V(x_0, y_0) \subset D$. Seuls sont considérés les points intérieurs de D : le domaine de I' est alors $D' = B$.

-Le voisinage utilisé est un voisinage tronqué $V \cap D$ et $D' = D$.

Dans ce cas, c'est la Topologie liée au sous ensemble D qui est utilisée.

D'une manière générale, étant donné un objet A de l'image I (restriction de I à un sous ensemble A de D), la transformation locale est optimale pour \hat{A} .

Pour les points du bord de A, la valeur transformée prend en compte les tons de gris du point n'appartenant pas à A; faussant ainsi le resultat.

REMARQUE:

Ceci met en valeur l'importance du rapport de la taille du voisinage utilisé et de la taille des objets de l'image. Un voisinage trop grand donne : $\sqrt{a/A} \quad b/A / b \quad \sqrt{a}$
(le voisinage dépasse l'objet A) d'où $\hat{A} = \emptyset$. Le resultat est mauvais.

2.3. DESCRIPTION DES TRANSFORMATIONS LOCALES UTILISEES:

L'operateur K de la transformation locale est souvent lineaire. Dans ce cas, la valeur de l'image transformée en un point est une combinaison lineaire des tons de gris de ce point et de ses voisins.

La transformation est alors définie d'une part par la taille du voisinage et d'autres part, par la matrice carré de (a_{kl}) . La taille du voisinage est liée à la fréquence spatiale des phénomènes étudiés, la matrice (a_{kl}) met en valeur une propriété particulière de la distribution des tons de gris sur le voisinage.

Ces transformations décrivent le comportement de la fonction $I(x,y)$ au voisinage du point (x,y) . Celles-ci sont souvent adaptées par la discrétisation des opérateurs de géométrie différentielle (gradient, Laplacien).

Il existe donc, d'une part des operateurs de differentiation (1^{er} de 2^{eme} ordre) qui mettent en valeur des contrastes et des variations de tons de gris, et d'autres part, des operateurs d'integration qui aboutissent à une régularisation de l'image.

Nous nous interesserons dans ce qui suit à des transformations locales sur des voisinages V_1 .

2.3.1. MOYENNE:

C'est un opérateur d'integration qui calcule la moyenne arithmétique des tons de gris du point et de ses voisins.

$$A_{\text{moy}} = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \quad I'_{\text{moy}}(x_0, y_0) = \frac{1}{9} \left[\sum_{k=-1}^1 \sum_{l=-1}^1 I(x_0+k, y_0+l) \right]$$

2.3.2. ECART A LA MOYENNE:

C'est un opérateur de différentiation du 2^{ème} ordre correspondant au Laplacien (Rosenfield 69). Il calcule la différence entre le ton de gris du point central et la moyenne des tons de gris du voisinage.

$$A_{\text{ec}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - A_{\text{moy}} = \begin{bmatrix} -1/9 & -1/9 & -1/9 \\ -1/9 & +8/9 & -1/9 \\ -1/9 & -1/9 & -1/9 \end{bmatrix}$$

L'écart à la moyenne met en valeur la singularité du point par rapport à son voisinage, il peut être positif ou négatif. Dans ce dernier cas, un décalage de l'origine est nécessaire pour satisfaire la définition d'une image numérisée $(I(x, y), N)$.

2.3.3. ECART TYPE SUR LE VOISINAGE:

$$I'(x_0, y_0) = \left(\frac{1}{9} \cdot \sum_{k=-1}^1 \sum_{l=-1}^1 I^2(x_0+k, y_0+l) - \left(\frac{1}{9} \cdot \sum_{k=-1}^1 \sum_{l=-1}^1 I(x_0+k, y_0+l) \right)^2 \right)^{1/2}$$

Cet opérateur est utilisé comme mesure d'homogénéité de la distribution des tons de gris sur le voisinage. Contrairement à l'écart à la moyenne, il ne privilégie pas le point central.

2.3.4. AUGMENTATION DU CONTRASTE:

Le ton de gris initial est remplacé par sa valeur augmentée algébriquement de l'écart à la moyenne.

Un point plus clair (respectivement plus foncé) que ses voisins deviendra encore plus clair (respectivement plus foncé).

$$A_c = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + A_{\text{ec}} = \begin{bmatrix} -1/9 & -1/9 & -1/9 \\ -1/9 & +17/9 & -1/9 \\ -1/9 & -1/9 & -1/9 \end{bmatrix}$$

$$I'_c(x_0, y_0) = I(x_0, y_0) + I'_{\text{ec}}(x_0, y_0)$$

2.4. PRESENTATION DES RESULTATS:

2.4.1. ORGANIGRAMME ET PROGRAMMES:

2.4.1.1. METHODE DE ROSENFELD ET AL (71,72):

Cette méthode comprend quatre étapes dont l'organigramme est schématisé par la figure.5.

-Fonction des sous-programmes utilisés:

1^{ère} étape:

S/P INITIA son rôle est de ranger les données sur les différents vecteurs nécessaire aux calculs.

S/P MOY1 son rôle est de calculer la moyenne de tous les points d'une fenêtre (2x2).

S/P MOY2 son rôle est de prendre une fenêtre 4x4 et de calculer la moyenne de ses points.

S/P MOY3 même rôle que MOY1 et MOY2.

Fenetre prise : 8x8 .

S/P RANG1 sa fonction est d'écrire les différentes moyennes sur disque.

2^{ème} étape:

S/P CLASS1, CLASS2, CLASS3 ils sont conçus pour classer les différentes moyennes sur des vecteurs permettant ainsi le calcul des différences.

S/P DIF1, DIF2, DIF3 leur fonction est de calculer les différences dans toutes les directions (DIF1: pour la taille 1, DIF2: pour la taille 2, DIF3: pour la taille 3).

S/P SUIT1, SUIT2, SUIT3 ils servent à transmettre le contenu de la ligne i-dans la ligne i-1 afin de balayer toute l'image.

S/P RANG1, RANG2: leur rôle est d'écrire respectivement la différence et la meilleure orientation sur disque.

3^{ème} étape:

S/P RANG3: son rôle est d'écrire la meilleure taille sur disque. (les S/P RANG1 et RANG2 ont la même fonction que dans la 2^{ème} étape).

4^{ème} étape:

S/P ORDRE: il a la même fonction que le S/P INITIA.

S/P TRACE1, TRACE2, TRACE3: leur rôle est de faire respectivement

le traitement de la première, deuxième et troisième ligne de l'image.

S/P TRACE: sa fonction est de tracer les frontières du reste de l'image.

S/P SUIT: il a le même rôle que SUIT1, SUIT2 et SUIT3.

2.4.1.2. DIFFERENTES METHODES DES T.L :

- L'organigramme est schématisé par la figure 6.
- Fonctions des sous programmes utilisés:

S/P MOYENN: son rôle est de calculer la moyenne de tous les points pris dans une fenêtre 3x3, ainsi qu'il est utilisé pour le calcul du contraste, l'écart à la moyenne et l'écart type.

Notons que les S/P INITIA et SUIT ont les mêmes fonctions que précédemment.

fig.5

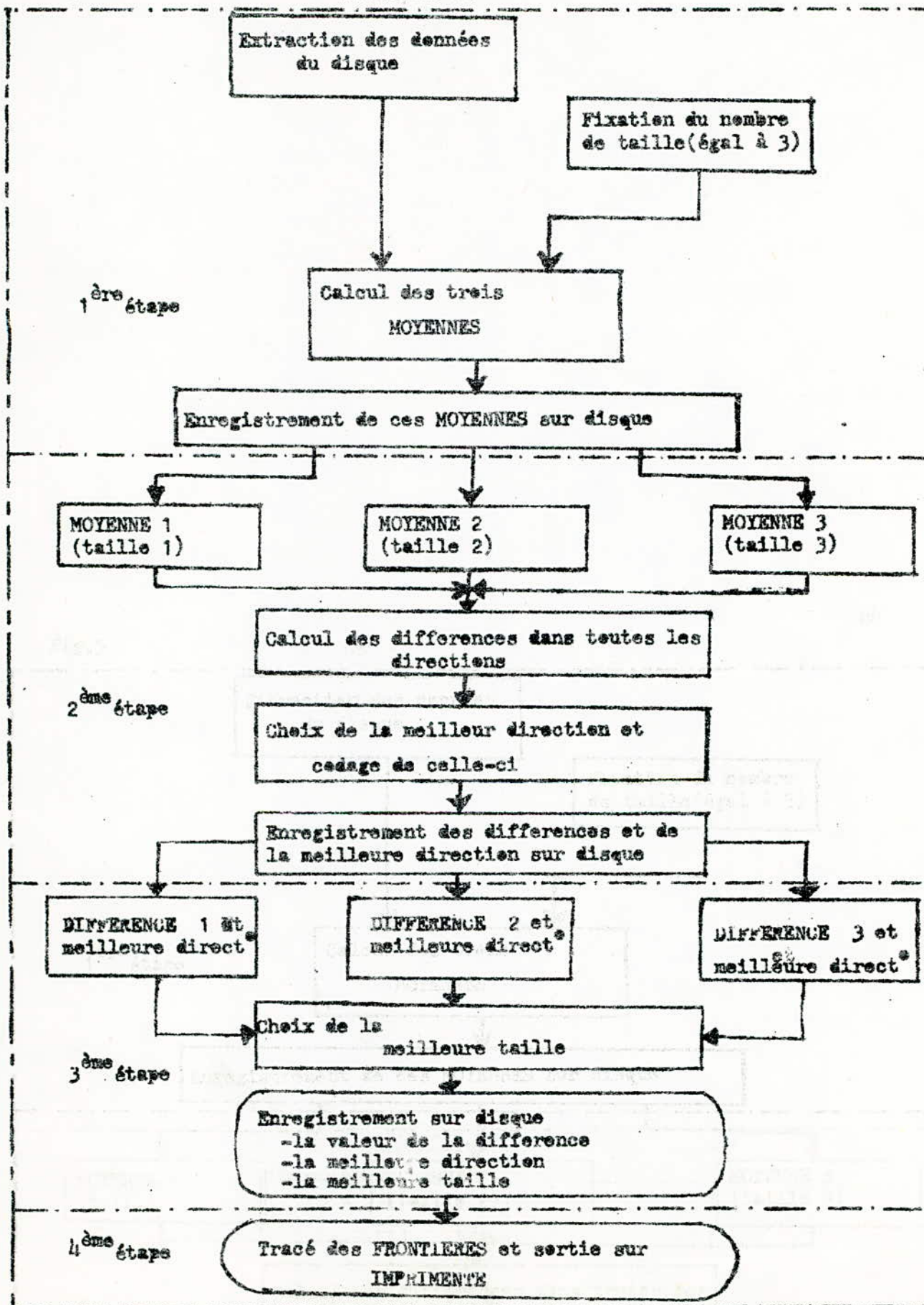
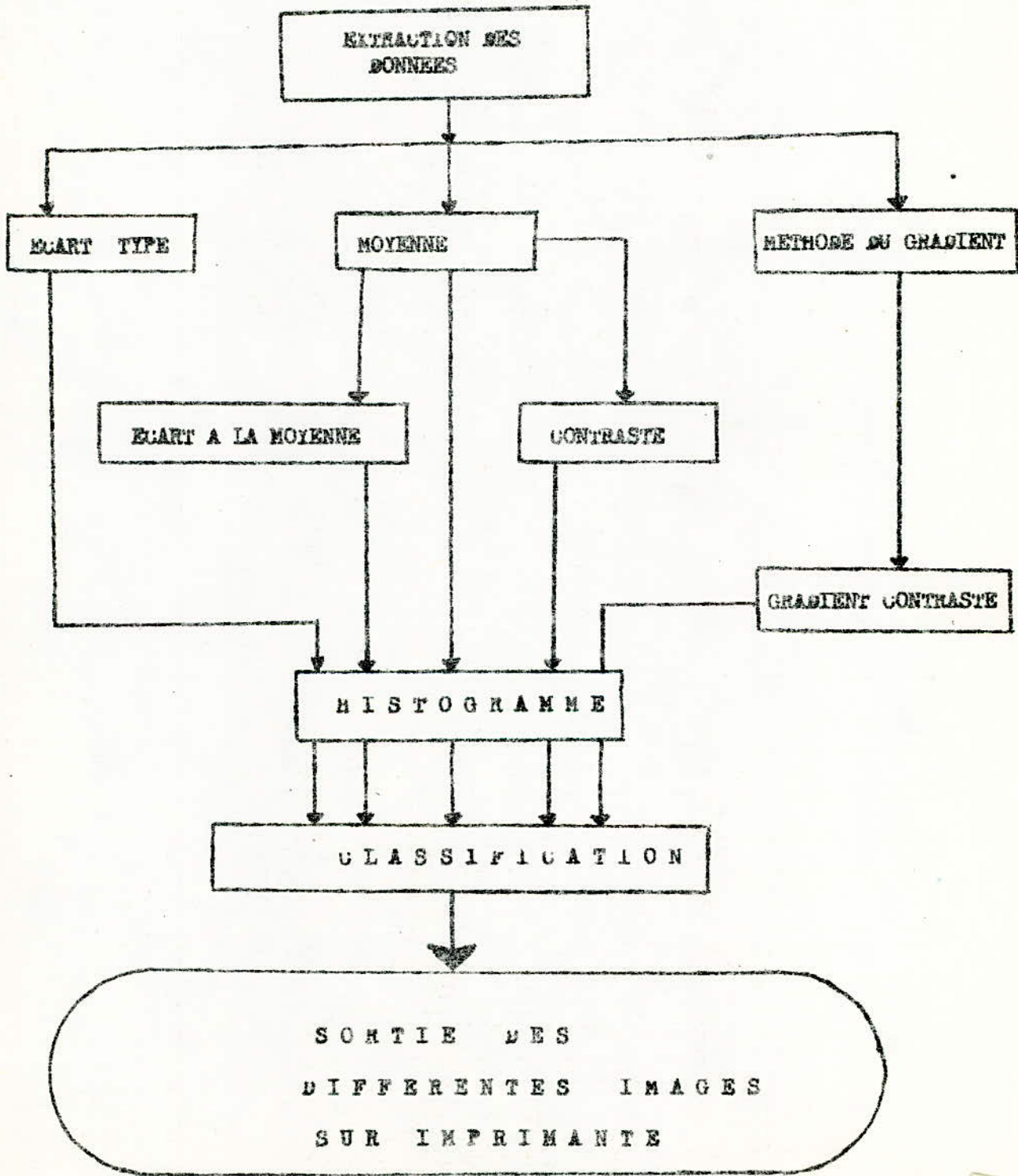
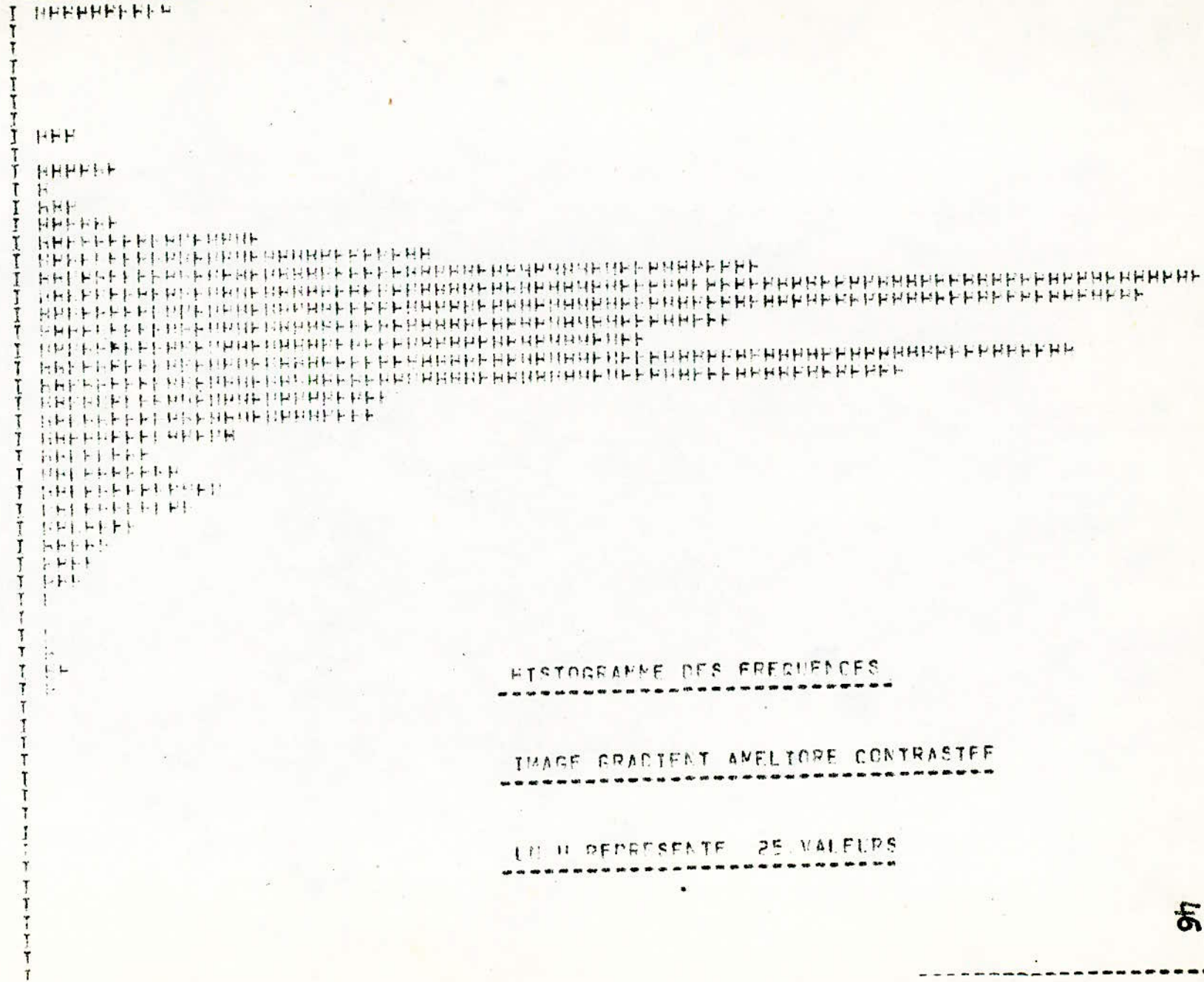


fig.6



NIVEAU FREQUENCE

1 252
 2 0
 3 0
 4 0
 5 1
 6 4
 7 7
 8 77
 9 5
 10 150
 11 21
 12 90
 13 157
 14 306
 15 707
 16 1200
 17 2058
 18 1672
 19 1320
 20 1007
 21 1827
 22 1581
 23 631
 24 616
 25 374
 26 200
 27 252
 28 220
 29 276
 30 178
 31 120
 32 113
 33 79
 34 32
 35 24
 36 28
 37 28
 38 55
 39 41
 40 11
 41 7
 42 8
 43 12
 44 5
 45 7
 46 4
 47 1
 48 3
 49 2
 50 4
 51 4
 52 1
 53 1
 54 1
 55 1



HISTOGRAMME DES FREQUENCES

IMAGE GRADIENT AMELIORE CONTRASTE

UN U REPRESENTE 25 VALEURS

I M A G E D ' U N E

R E G I O N

D E L ' I N D E

{ P A L N I }

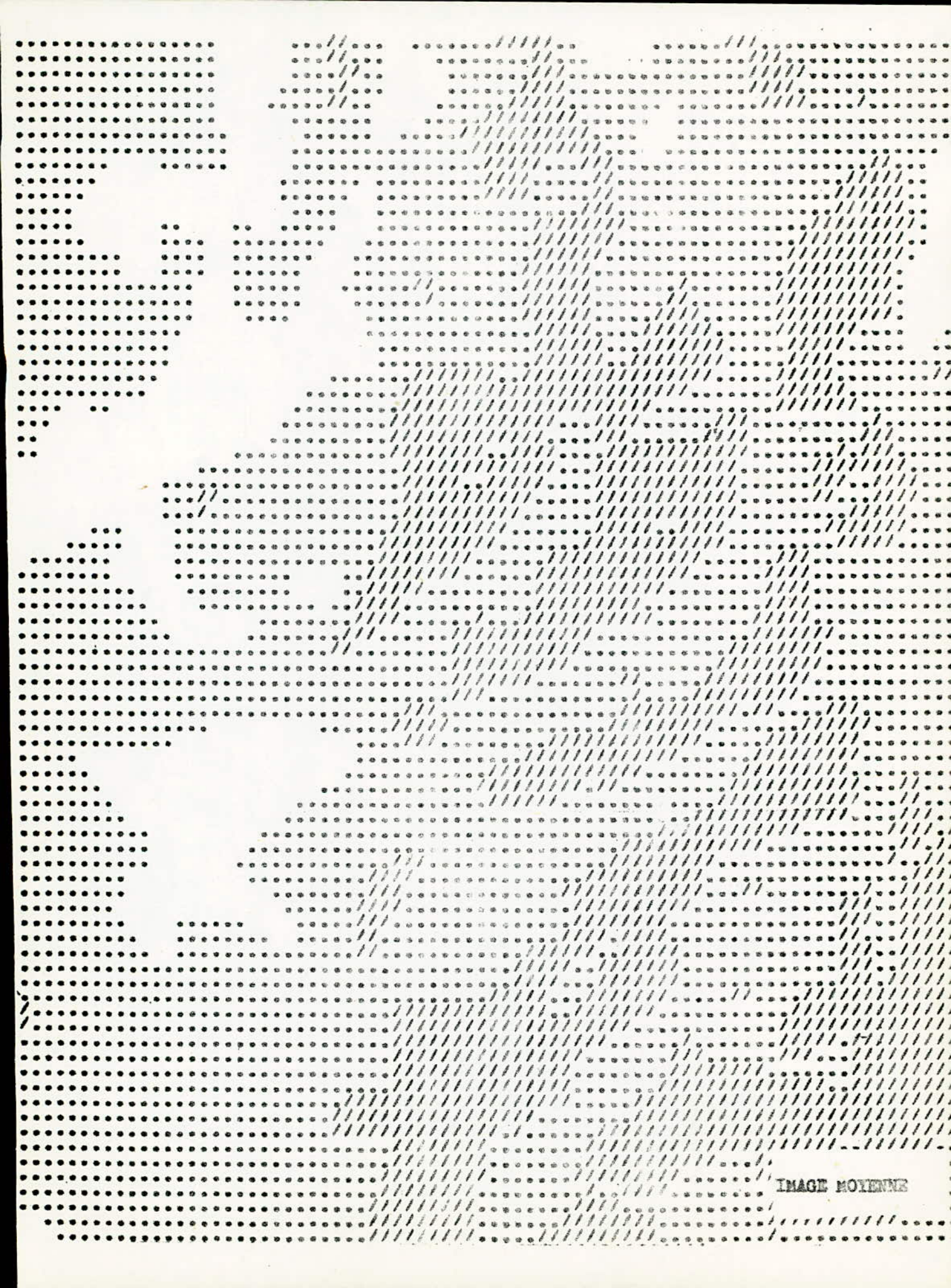


IMAGE MOYENNE



IMAGE CONTRAST

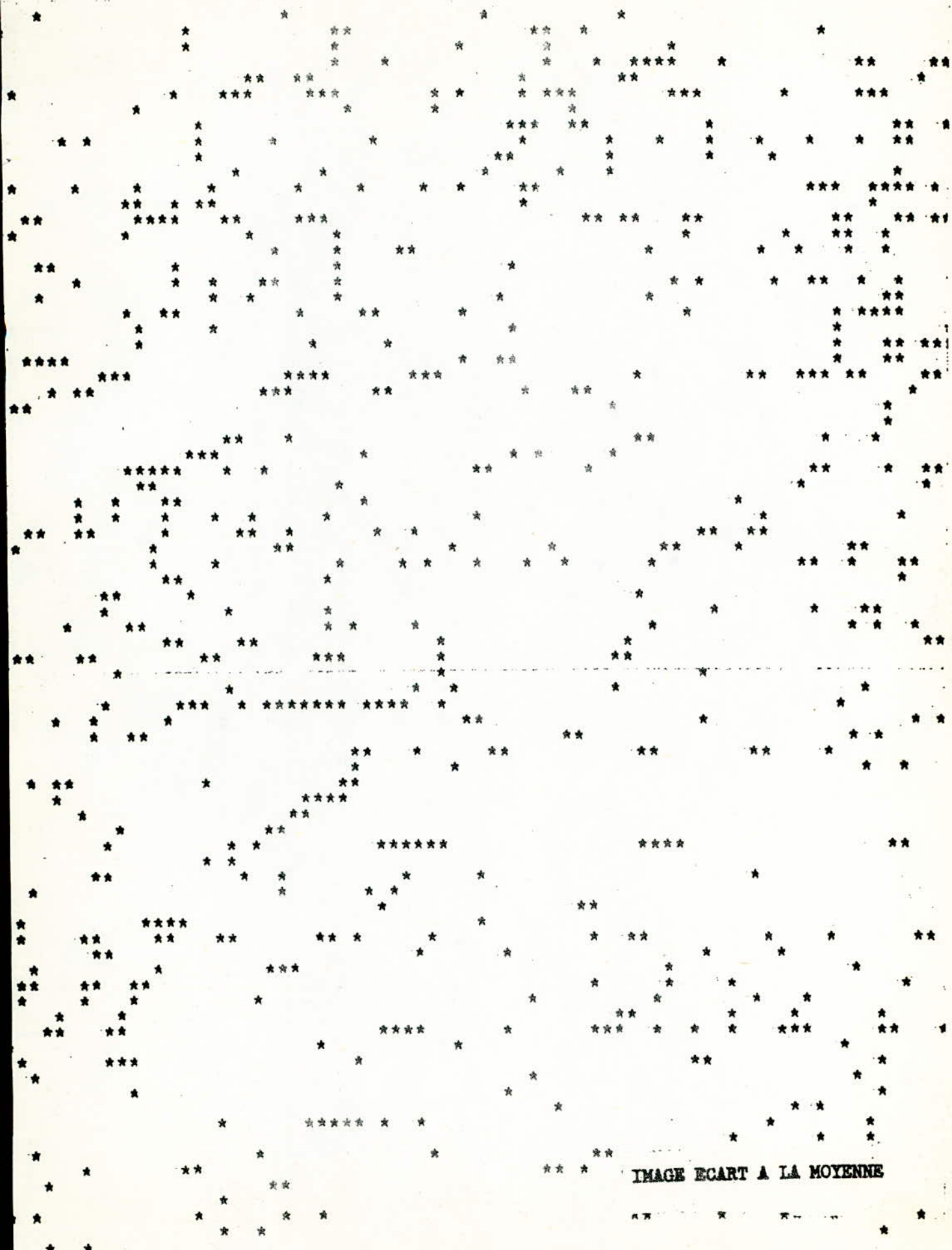


IMAGE ECART A LA MOYENNE

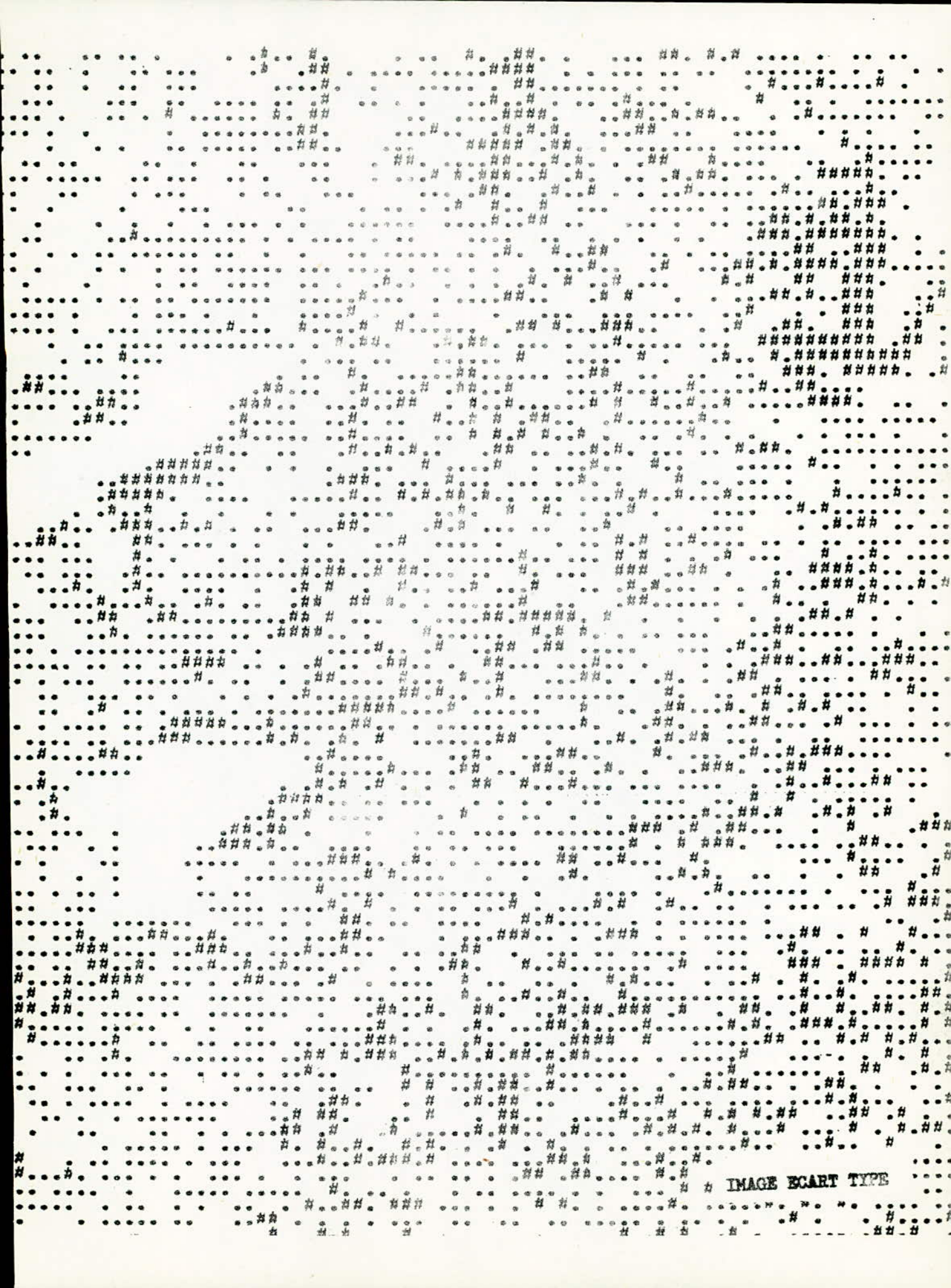


IMAGE SCART TYPE



IMAGE GRADIENT AMELIORE

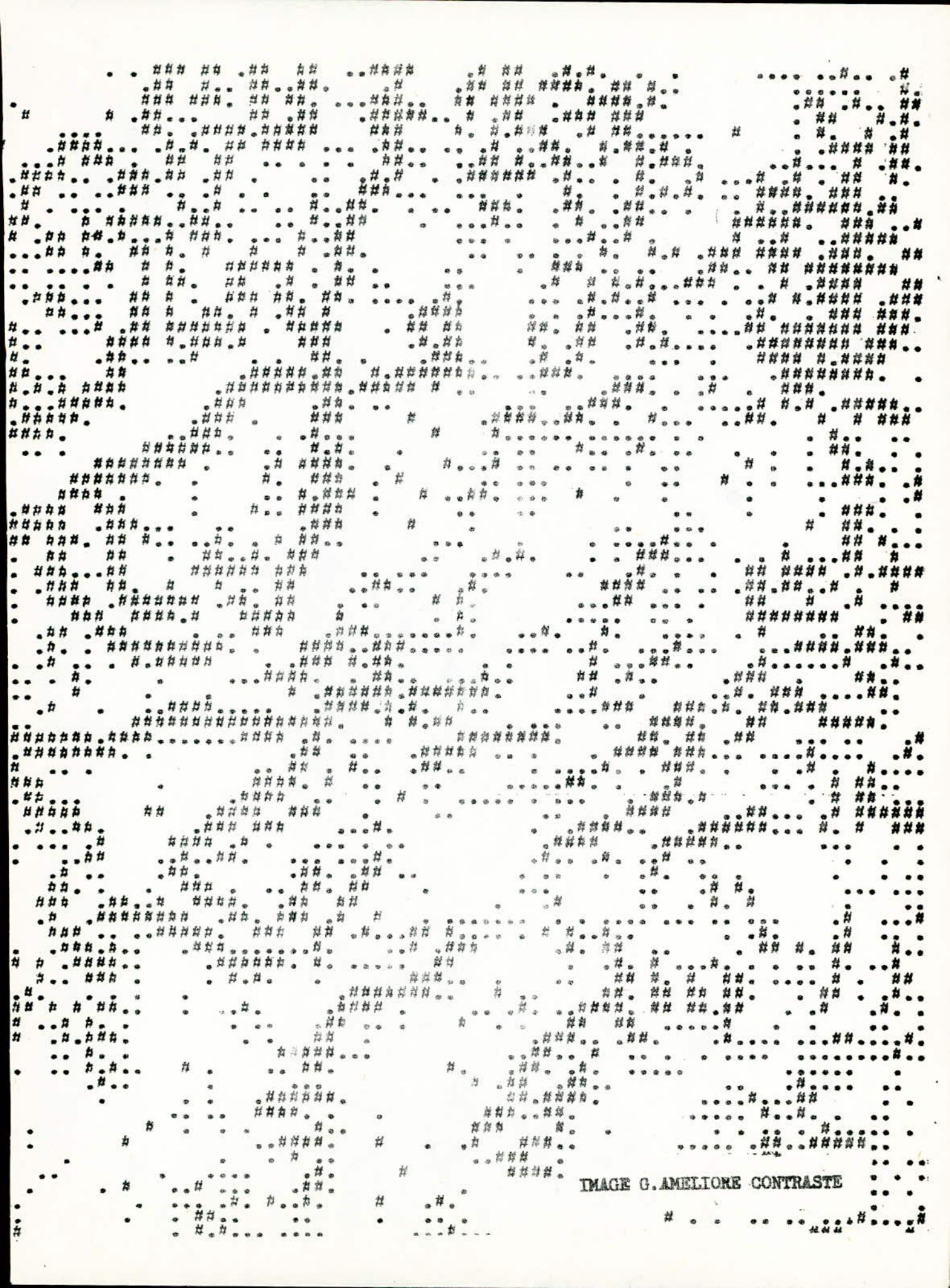


IMAGE G. AMELIOME CONTRASTE

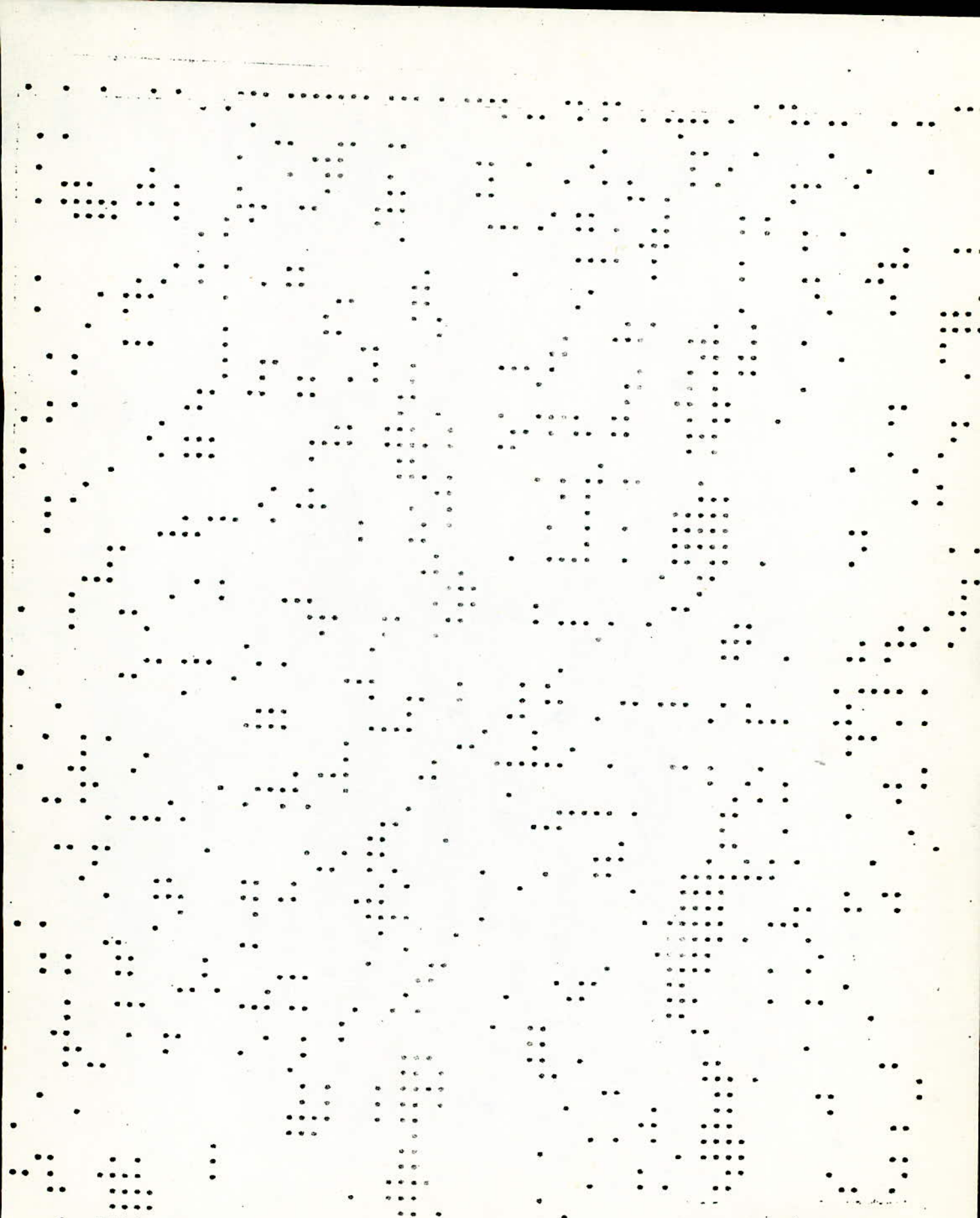


IMAGE ROSENFELD ET AL

I M A G E D ' U N E

R E G I O N

D ' A L G E R

{ B A I E D ' A L G E R }

IMAGE MOYENNE



IMAGE G. AMELIORE CONTRASTE

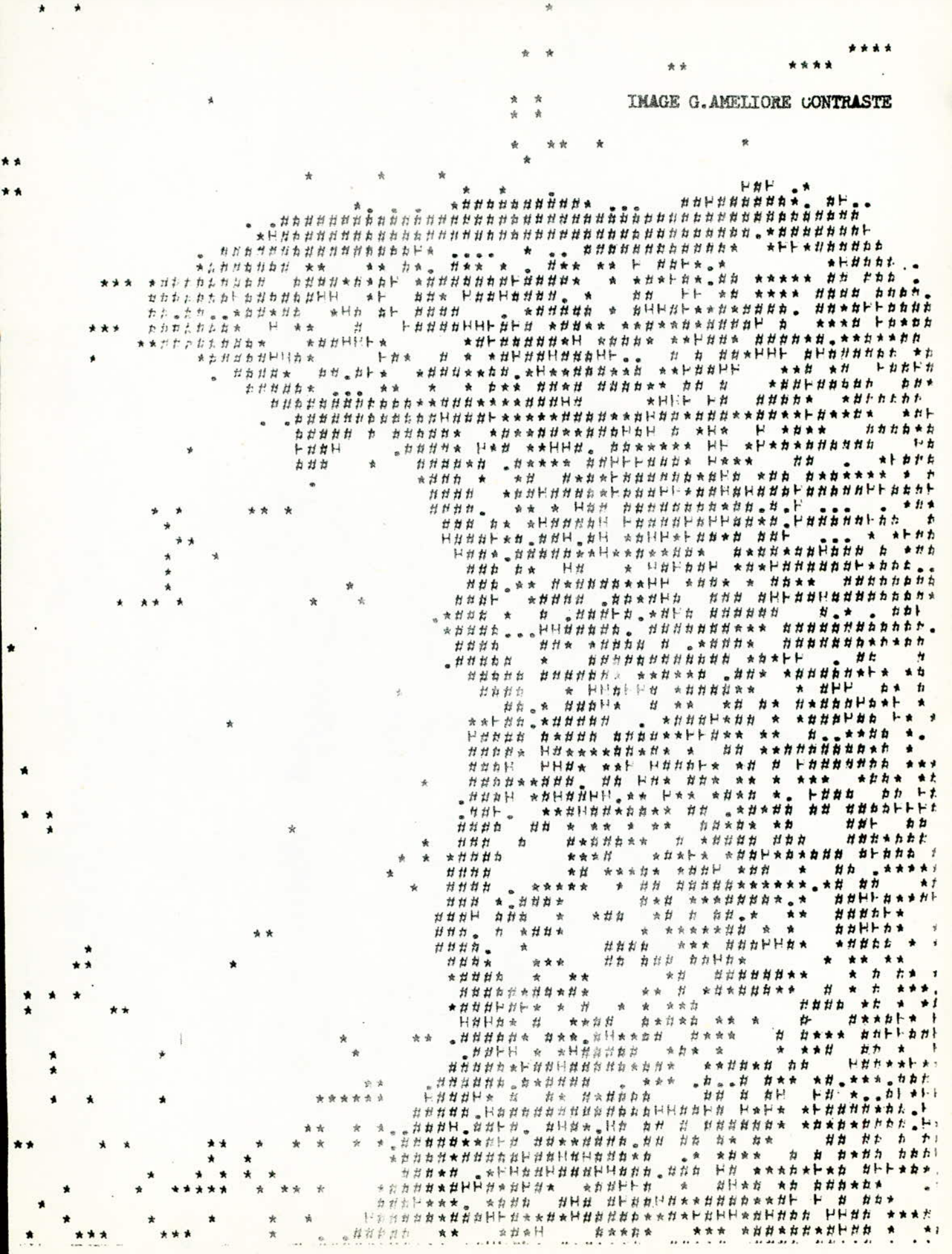
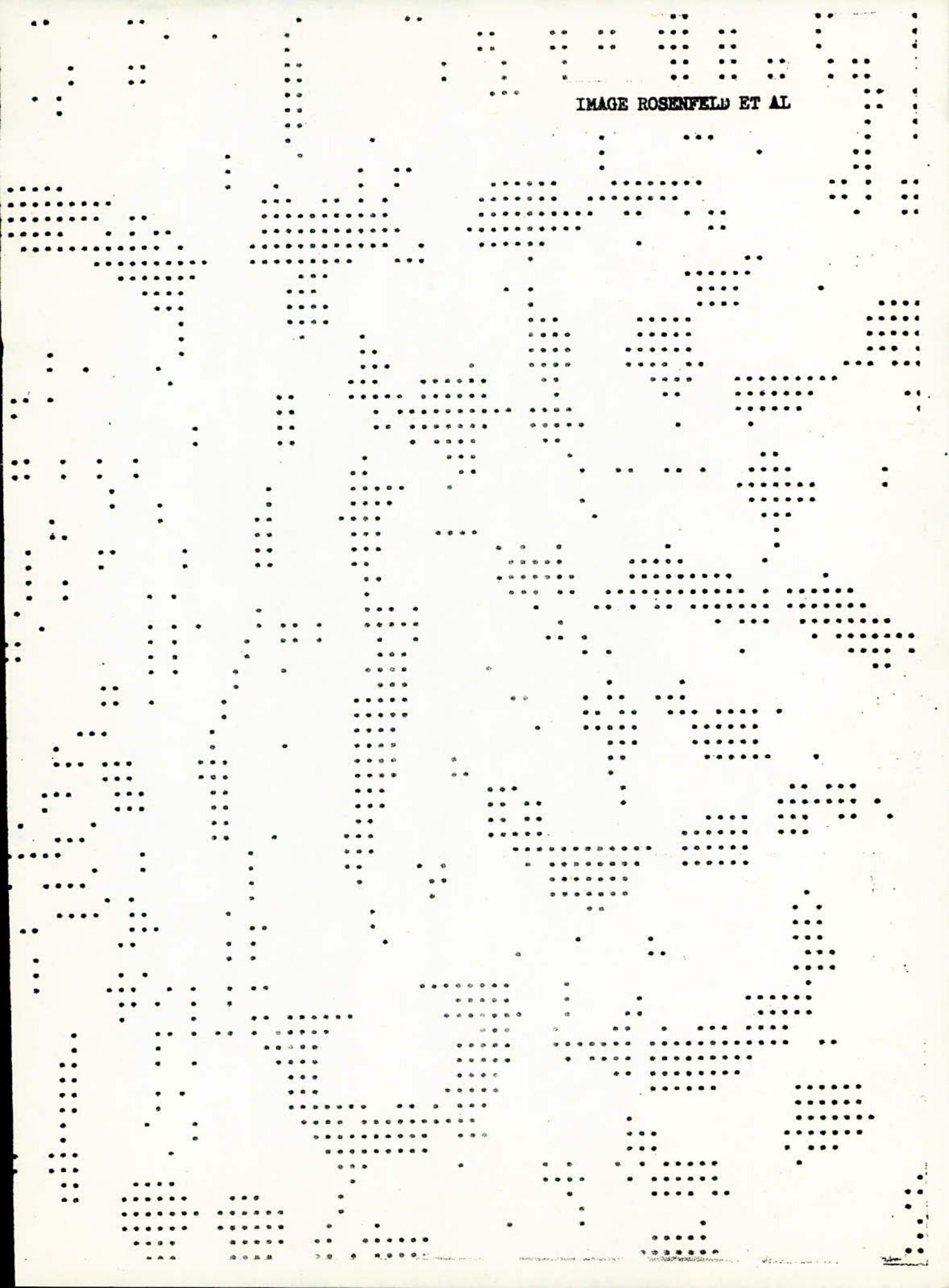


IMAGE ECART TYPE



IMAGE ROSENFELD ET AL



2.4.2. COMMENTAIRES DES RESULTATS:

Les zones étudiées sont extraites d'images LANDSAT, la première correspond à une région de l'INDE, la seconde à une région d'ALGER (baie d'alger).

Les résultats présentés portent sur une sous image (128x128) points.

Les deux régions étudiées sont extraites de l'image du satellite LANDSAT 1, celle d'alger a été prise le 30 Septembre 1979.

Plusieurs transformations locales sont appliquées à une partie de l'image M.S.S.7 (canal 7) dont la bande spectrale est de 0,8-1,1 micromètre.

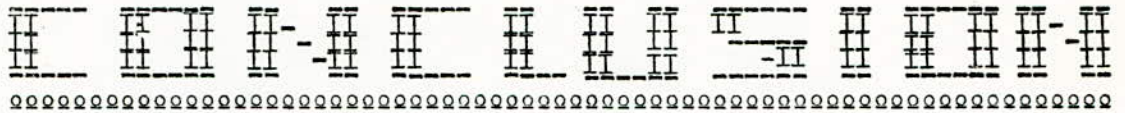
Sur l'image originale, le découpage de l'histogramme des tons de gris fait apparaître différentes classes.

Parmi les diverses transformées obtenues il y a celles qui présentent plus d'intérêt pour l'interpréteur par rapport à d'autres, et cela n'est dû qu'au choix du but recherché.

Nous citons l'effet de l'utilisation de chaque transformation locale appliquées sur les deux images indiquées ci-dessus, et cela par un tableau de la figure.7 dont lequel on résume toutes les transformations locales utilisées ainsi que le temps d'exécution des calculs mis par l'ordinateur pour chaque T.L. sur une image de (128x128) points

L'IMAGE QUE NOUS AVONS TRAITÉ EST DE DIMENSION (128x128) POINTS.

NOI DE LA METHODE UTILISEE	L'EFFET DE LA METHODE SUR L'IMAGE	OBSERVATIONS	TEMPS DE CALCUL	TEMPS DE SORTIE D'INSTRUMENTATION	TEMPS DE SORTIE DE L'IMAGE	TEMPS DE SORTIE GLOBAL
MOYENNE	-la difference entre les tons de gris de points voisins est atténuée -l'homogénéité est accentuée -le contraste est diminué.	les grandes zones homogènes sont mis en évidence.	31" 7	27"	38" 5	1' 37" 2
CONTRASTE	-le contraste est plus accentué	l'amélioration principale est une mise en évidence des structures.	34" 6	33" 7	39" 7	1' 48" 2
ECART A LA MOYENNE	-les points de forts contraste sont détectés.	cette détection dépend de la taille du voisinage.	47" 2	32" 7	37" 1	1' 57"
ECART TYPE	-permet de mesurer le degré d'homogénéité	régularisation des résultats	54" 8	38"	42"	2' 14" 8
GRADIENT AMELIORE	-les frontières sont détectées	dépend de ce qu'on veut détecter.	53" 5	40" 7	26" 7	2' 00" 9
GRADIENT AMELIORE CONTRASTE	-permet d'accentuer le contraste des frontières détectées		43" 6	44" 9	31"	1' 59" 5
ROSENFELD et AL	-permet de détecter toutes les frontières existantes sur la zone étudiée.	engorgement des frontières, difficulté d'interprétation		1 ^{ère} étape: 2' 12" 7 2 ^{ème} étape: 1' 15" 6 3 ^{ème} étape: 1' 19" 8 4 ^{ème} étape: 2' 04" 5		6' 55" 6



Après avoir défini le concept de reconnaissance de formes et de texture, utilisé dans l'étude des images de télédétection, nous avons développé les différentes possibilités d'utilisation des transformations locales pour l'étude des informations texturales d'une image.

L'étude de ces possibilités montre que la création d'un répertoire ou dictionnaire de transformations locales ainsi que d'une syntaxe formée d'un ensemble de méthodes qui associent ces transformations entre elles permettra de développer un langage descriptif de la texture dont les avantages principaux sont la souplesse et la simplicité d'implantation informatique.

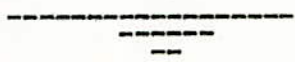
Appliquées aux images de télédétection, ces méthodes d'analyse de la texture permettent en particulier les réalisations suivantes :

- l'amélioration d'image en vue de l'interprétation visuelle, notamment pour le traitement en interactif. Cette amélioration porte sur la précision des contours, le tracé des frontières, etc... et s'applique à la fois sur des images aériennes et satellitaires.
- l'identification des classes d'objets au sol par leur texture.

Notre travail a été entravé par des problèmes, surtout du côté centre de calcul à cause notamment:

- du nombre important d'utilisateurs pour un petit centre de calcul.
- des pannes fréquentes et parfois prolongées du matériel.

Ils serait intéressant de compléter notre travail pour pouvoir étudier une image multispectrale complète et cela en ajoutant aux paramètres spectraux les paramètres texturaux de l'image suivis d'une classification multidimensionnelle, afin d'aboutir à un résultat très efficace en traitement numérique d'images.



- B I B L I O G R A P H I E -

- A. COUZY
"LA TELEDETECTION"
EDITION Que sais-je? 1981

- J. DESACHY
"CONTRIBUTION A L'ETUDE DE LA TEXTURE EN
TELEDETECTION" 1980

- M. ATEIA ABD EL-FATAH SHAHIN
"TRAITEMENT D'IMAGES RADAR EN VUE DE L'ETUDE DE SURFACES
AGRICOLES ET FORESTIERES PAR TELEDETECTION" 1980

- M. JACQUES QUACH
"UTILISATION DE TRANSFORMATIONS LOCALES POUR L'ETUDE DE
LA TEXTURE DES IMAGES DE TELEDETECTION" 1979

- M. TILIKETE, M. BELKACEMI
"CONCEPTION ET REALISATION D'UN LOGICIEL POUR
TRAITEMENT ET MANIPULATION D'IMAGES SPATIALES" Janvier82

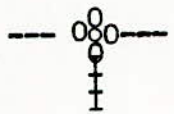
- M. DREYFUS
"FORTRAN IV"
edition DUNOD 1968

- P. CASSIRAME
"METHODES DE CLASSIFICATION DES DONNEES MULTIDIMENSIONNELLES
EN TELEDETECTION. APPLICATION A LA CARTOGRAPHIE D'UNE
REGION AGRICOLE (CAMARQUE)" 1977

- J. M. ROMEDER
"METHODES ET PROGRAMMES D'ANALYSE DISCRIMINANTE"
edition DUNOD 1973

```
      A      N N N N EEEEE X X EEEEE
     A A A A N N N N EEE   X X EEE
    A A A A A A N N N N EEE   X X EEE
   A A A A A A A N N N EEEEE X X EEEEE
```

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
```



P R O G R A M M E S

U T I L I S E S

D A N S

C E T T E

M E M O I R E

1.

I- Gradient simple

II- Gradient amélioré

III- Filtrage et lissage des lignes de contours
après l'opération gradient amélioré

2.

IV-les quatre etapes de la methode de ROSENFELD et AL

3.

V- Les transformations locales:

-Moyenne

-Ecart à la moyenne

-Contraste

-Ecart type

-Gradient amélioré

-Contraste du gradient amélioré

-Sortie d'histogramme

-Classification et sortie d'image

P R O G R A M M E S

C O N C E R N A N T

L E S

A L G O R I T H M E S

S I M P L E S D E

G R A D I E N T S I M P L E

E T

G R A D I E N T A M E L I O R E


```

C      PROGRAMME DU GRADIENT SIMPLE
C      DONNANT UNE IMAGE CODEE APRES SEUILLAGE
DIMENSION MAT(30,60),MA(30,60),MT(30,60)
DATA IPLANC,IPCINT/1H,1H./
READ(105,100)((MAT(I,J),J=1,60),I=1,30)
READ(105,101)(N,M)
IA=N-1
IP=M-1
DO 40 I=1,IA
DO 50 J=1,IP
MA(I,J)=IABS(MAT(I,J)-MAT(I+1,J+1))+IABS(MAT(I,J+1)-MAT(I+1,J))
50 CONTINUE
MA(J,J)=MA(I,J-1)
40 CONTINUE
DO 55 J=1,M
MA(N,J)=MA(IA,J)
55 CONTINUE
DO 60 I=1,N
DO 70 J=1,M
IF(MA(I,J).GE.P) GO TO 6
MT(I,J)=IPLANC
GOTO 70
6 MT(J,J)=IPCINT
70 CONTINUE
60 CONTINUE
PRINT 200
PRINT 190,((MAT(I,J),J=1,N),I=1,N)
PRINT 165
PRINT 166
PRINT 160
PRINT 161
PRINT 150,((MA(K,L),L=1,N),K=1,N)
PRINT 160
PRINT 162
PRINT 170,((MT(I,J),J=1,N),I=1,N)
100 FORMAT(40I2)
101 FORMAT(2I2)
150 FORMAT(4X,60I2)
160 FORMAT(1H1)
161 FORMAT(10X,'IMAGE GRADIENT SIMPLE',/////)
162 FORMAT(15X,'IMAGE CODEE PAR DES BLANCS ET DES POINTS',/////)
165 FORMAT(1X,/////)
166 FORMAT(100X,'IMAGE DIGITALISEE')
170 FORMAT(5X,60A1)
190 FORMAT(4X,60I2)
200 FORMAT(10X,'MATRICE DE DONNEES',/////)
STOP
END

```

PROGRAMME DU GRADIENT AMELIORE

IMAGE CODEE APRES SEUILLAGE

```

DIMENSION MAT(30,60),MA(30,60),MT(30,60)
DATA IRLANC,IPCINT/1H,1F/
READ(105,100)((MAT(I,J),J=1,60),I=1,30)
READ(105,101)(N,M)
IA=N-3
IR=M-3
DO 40 I=1,IA
DO 50 J=1,IR
IC=IABS(MAT(I+1,J+1)-MAT(I+2,J+2))+IABS(MAT(I+2,J+1)-MAT(I+1,J+2))
ID=IABS(MAT(I,J)-MAT(I+3,J+3))+IABS(MAT(I+3,J)-MAT(I,J+3))
IW=IABS(MAT(I,J+1)-MAT(I+3,J+2))+IABS(MAT(I+2,J)-MAT(I+1,J+3))
IZ=IABS(MAT(I,J+2)-MAT(I+3,J+1))+IABS(MAT(I+1,J)-MAT(I+2,J+3))
IM1=SGRT(IC*ID)
IM2=SGRT(IW*IZ)
IM3=SQRT(IM1)
IM4=SGRT(IM2)
MA(I,J)=IM3*IM4
CONTINUE
CONTINUE
DO 41 I=1,29
DO 51 J=58,59
MA(I,J)=IABS(MAT(I,J)-MAT(I+1,J+1))+IABS(MAT(I,J+1)-MAT(I+1,J))
CONTINUE
MA(I,J)=MA(I,J-1)
CONTINUE
DO 42 I=28,29
DO 52 J=1,57
MA(I,J)=IABS(MAT(I,J)-MAT(I+1,J+1))+IABS(MAT(I,J+1)-MAT(I+1,J))
CONTINUE
CONTINUE
DO 53 J=1,60
MA(30,J)=MA(29,J)
CONTINUE
DO 60 I=1,30
DO 70 J=1,60
IF(MA(I,J).GE.9)GO TO 5
MT(I,J)=IRLANC
GOTO 70
5
70
60
MT(I,J)=IPCINT
CONTINUE
CONTINUE
PRINT 200
PRINT 190,((MAT(I,J),J=1,60),I=1,30)
PRINT 165
PRINT 166
PRINT 160
PRINT 161
PRINT 150,((MA(K,L),L=1,60),K=1,30)
PRINT 160
PRINT 162
PRINT 170,((MT(I,J),J=1,60),I=1,30)
100
101
150
161
162
165
166
170
190
200
160
FORMAT(40I2)
FORMAT(2I3)
FORMAT(4X,60I2)
FORMAT(10X,'IMAGE GRADIENT AMELIORE',/////)
FORMAT(15X,'IMAGE CODEE PAR DES BLANCS ET DES POINTS',/////)
FORMAT(1X,/////)
FORMAT(100X,'IMAGE DIGITALISEE')
FORMAT(5X,60A1)
FORMAT(4X,60I2)
FORMAT(10X,'MATRICE DE DONNEES',/////)
PRINT 160
FORMAT(1F1)
STOP
END

```



```

DIMENSION MAT(30,60),MB(30,60),MA(32,62),MN(32,62),M(32,62)
DATAIRLANS,INCINS/1H',1H-/
READ(105,100)((MAT(I,J),J=1,60),I=1,30)
READ(105,101)(N,L)
PRINT170
PRINT120
PRINT130
PRINT110,((MAT(I,J),J=1,60),I=1,30)
CALL GRADA(MAT,MB,N,L)
CALL IPIA(MB)
CALL TRANS(MB,MA)
CALL ILISS(MA,MN)
CALL ICCD(MN,M)
100  FORMAT(40I2)
101  FORMAT(2I2)
170  FORMAT(1H1)
110  FORMAT(4X,60I2)
120  FORMAT(4X,'DEFLECTION DE CONTOURS',///// )
130  FORMAT(50X,'MATRICE DE DONNEES',///// )
STOP
END

```

```

SUBROUTINE TRANS(MB,MA)
DIMENSION MB(30,60),MA(32,62)
DO60I=1,32
DO70J=1,62
MA(I,J)=0
70  CONTINUE
60  CONTINUE
K=0
DO50I=2,31
K=K+1
L=0
DO51J=2,61
L=L+1
51  MA(I,J)=MB(K,L)
50  CONTINUE
PRINT170
PRINT200
PRINT120,((MA(I,J),J=1,62),I=1,32)
170  FORMAT(1H1)
200  FORMAT(50X,'IMAGE TRANSFORMEE',///// )
120  FORMAT(4X,62I1)
RETURN
END

```

```

SUBROUTINE ICCD(MN,M)
DIMENSION MN(32,62),M(32,62)
DATAIRLANS,INCINS/1H',1H-/
DO10I=1,32
DO20J=1,62
M(I,J)=IRLANS
20  CONTINUE
10  CONTINUE
DO82I=1,32
DO92J=1,62
IF(MN(I,J).EQ.0)GOTO86
M(I,J)=INCINS
GOTO92
86  M(I,J)=IRLANS
92  CONTINUE
92  CONTINUE
PRINT170
PRINT200
PRINT180,((M(I,J),J=1,62),I=1,32)
170  FORMAT(1H1)
200  FORMAT(50X,'IMAGE CODEE AVEC DES BLANCS ET DES TITRES',///// )
180  FORMAT(4X,62A1)
RETURN
END

```

```

SUBROUTINE GRADA(MAT,MB,N,I)
DIMENSION MAT(30,60),MR(30,60)
IA=I-3
IB=I-2
DO 40 J=1,IA
DO 50 J=1,IB
IC=IABS(MAT(I+1,J+1)-MAT(I+2,J+2))+IABS(MAT(I+2,J+1)-MAT(I+1,J+2))
ID=IABS(MAT(I,J)-MAT(I+3,J+3))+IABS(MAT(I+3,J)-MAT(I,J+3))
IE=IABS(MAT(I,I+1)-MAT(I+3,J+2))+IABS(MAT(I+2,J)-MAT(I+1,J+2))
IF=IABS(MAT(I,I+2)-MAT(I+3,J+1))+IABS(MAT(I+1,J)-MAT(I+2,J+3))
MR(I,J)=(IC+ID+IE+IF)**(0.25)
CONTINUE
40 CONTINUE
DO 51 J=1,IC
DO 51 J=1,IC
IE(I,J)=IABS(MAT(I,J)-MAT(I+1,J+1))+IABS(MAT(I,I+1)-MAT(I+1,J))
CONTINUE
51 CONTINUE
MR(I,J)=M(I,J-1)
CONTINUE
41 CONTINUE
DO 52 J=1,IC
DO 52 J=1,IC
IF(I,J)=IABS(MAT(I,J)-MAT(I+1,J+1))+IABS(MAT(I,J+1)-MAT(I+1,J))
CONTINUE
52 CONTINUE
42 CONTINUE
DO 53 J=1,60
MR(30,J)=MR(29,J)
CONTINUE
53 CONTINUE
PRINT 170
PRINT 170
PRINT 110,((MR(I,J),J=1,60),I=1,30)
170 FORMAT(1F11)
170 FORMAT(50X,'TRACE GRADIENT AMELIORE',//////)
110 FORMAT(4Y,60I2)
RETURN
END

```

```

SUBROUTINE TRIA(MB)
DIMENSION MB(30,60)
DO 120 I=1,30,5
DO 121 J=1,60,5
NR=0
K=0
124 AX=I+3
I=0
123 I=J+1
NR=NR+MB(I,I)
IF(L,00,4)GOTO122
L=L+1
GOTO123
122 IF(K,00,4)GOTO125
K=K+1
GOTO124
125 AM=NR/25
AM=AM+1
Y=0
132 AK=I+K
L=0
130 AI=I+L
IF(Y,00,4).GT.AM)GOTO128
NR(K,AI)=0
GOTO129
128 NR(K,AI)=1
120 IF(L,00,4)GOTO131
L=L+1
GOTO130
131 IF(L,00,4)GOTO121
K=K+1
GOTO122
121 CONTINUE
CONTINUE
PRINT 170
PRINT 200
PRINT 110,((MR(I,J),J=1,60),I=1,30)
170 FORMAT(1F11)
200 FORMAT(50X,'TRACE FILTRER',//////)
110 FORMAT(4Y,60I1)
RETURN
END

```



```

SUBROUTINE JLISS(MA,MN)
DIMENSION MA(32,62),MN(32,62)
DO 1 I=2,31
DO 91 J=2,61
IF (MA(I,J).EQ.0)GOTO1
IF (MA(I-1,J-1).EQ.0)GOTO26
IF (MA(I-1,J).EQ.0)GOTO19
IF (MA(I-1,J+1).EQ.0)GOTO10
IF (MA(I,J-1).EQ.0)GOTO4
IF (MA(I,J+1).EQ.0)GOTO3
IF (MA(I+1,J-1).EQ.0)GOTO1
IF (MA(I+1,J+1).EQ.0)GOTO1
3  MN(I,J)=1
   GOTO91
4  IF (MA(I,J+1).EQ.0)GOTO6
   IF (MA(I+1,J-1).EQ.0)GOTO1
   MN(I,J)=1
   GOTO91
6  IF (MA(I+1,J-1).EQ.0)GOTO7
   MN(I,J)=1
   GOTO91
7  IF (MA(I+1,J).EQ.0)GOTO8
   MN(I,J)=1
   GOTO91
8  IF (MA(I+1,J+1).EQ.0)GOTO1
   MN(I,J)=1
   GOTO91
10 IF (MA(I,J-1).EQ.0)GOTO16
   IF (MA(I,J+1).EQ.0)GOTO17
   IF (MA(I+1,J-1).EQ.0)GOTO1
   IF (MA(I+1,J).EQ.0)GOTO11
   MN(I,J)=0
   GOTO91
11 IF (MA(I+1,J+1).EQ.0)GOTO1
   MN(I,J)=1
   GOTO91
12 IF (MA(I+1,J).EQ.0)GOTO14
   MN(I,J)=0
   GOTO91
14 IF (MA(I+1,J+1).EQ.0)GOTO1
   MN(I,J)=1
   GOTO91
16 IF (MA(I,J+1).EQ.0)GOTO21
   IF (MA(I+1,J-1).EQ.0)GOTO1
   MN(I,J)=1
   GOTO91
18 IF (MA(I-1,J+1).EQ.0)GOTO20
   MN(I,J)=1
   GOTO91
20 IF (MA(I,J-1).EQ.0)GOTO21
   IF (MA(I,J+1).EQ.0)GOTO22
   IF (MA(I+1,J).EQ.0)GOTO21
   MN(I,J)=0
   GOTO91
22 IF (MA(I+1,J).EQ.0)GOTO23
   MN(I,J)=0
   GOTO91
23 IF (MA(I+1,J+1).EQ.0)GOTO1
   MN(I,J)=1
   GOTO91
26 IF (MA(I-1,J).EQ.0)GOTO50
   IF (MA(I-1,J+1).EQ.0)GOTO39
   IF (MA(I,J-1).EQ.0)GOTO32
   IF (MA(I,J+1).EQ.0)GOTO30
   IF (MA(I+1,J-1).EQ.0)GOTO1
   IF (MA(I+1,J).EQ.0)GOTO27
   MN(I,J)=0
   GOTO91
27 IF (MA(I+1,J+1).EQ.0)GOTO1
   MN(I,J)=1
   GOTO91
30 IF (MA(I+1,J+1).EQ.0)GOTO1
   MN(I,J)=1
   GOTO91

```

```

32 IF (MA(I, J+1).EQ.0) GOTO 35
   IF (MA(I+1, J-1).EQ.0) GOTO 1
   IF (MA(I+1, J).EQ.0) GOTO 21
   M(I, J)=0
   GOTO 91
35 IF (MA(I+1, J-1).EQ.0) GOTO 36
   M(I, J)=1
   GOTO 91
36 IF (MA(I+1, J).EQ.0) GOTO 37
   M(I, J)=1
   GOTO 91
37 IF (MA(I+1, J+1).EQ.0) GOTO 1
   M(I, J)=1
   GOTO 91
39 IF (MA(I, J-1).EQ.0) GOTO 46
   IF (MA(I, J+1).EQ.0) GOTO 43
   IF (MA(I+1, J-1).EQ.0) GOTO 1
   IF (MA(I+1, J).EQ.0) GOTO 40
   M(I, J)=0
   GOTO 91
40 IF (MA(I+1, J+1).EQ.0) GOTO 1
   M(I, J)=1
   GOTO 91
43 IF (MA(I+1, J).EQ.0) GOTO 44
   M(I, J)=0
   GOTO 91
44 IF (MA(I+1, J+1).EQ.0) GOTO 1
   M(I, J)=1
   GOTO 91
46 IF (MA(I, J+1).EQ.0) GO TO 21
   IF (MA(I+1, J-1).EQ.0) GO TO 1
   IF (MA(I+1, J).EQ.0) GO TO 21
   M(I, J)=0
   GOTO 91
50 IF (MA(I-1, J+1).EQ.0) GO TO 59
   IF (MA(I, J-1).EQ.0) GO TO 53
   IF (MA(I, J+1).EQ.0) GO TO 21
   IF (MA(J+1, J).EQ.0) GO TO 21
   M(I, J)=0
   GOTO 91
53 IF (MA(I, J+1).EQ.0) GO TO 21
   IF (MA(I+1, J-1).EQ.0) GO TO 55
   IF (MA(I+1, J).EQ.0) GO TO 21
   M(I, J)=0
   GOTO 91
55 IF (MA(I+1, J).EQ.0) GO TO 56
   M(I, J)=0
   GOTO 91
56 IF (MA(I+1, J+1).EQ.0) GO TO 21
   M(I, J)=0
   GOTO 91
59 IF (MA(I, J-1).EQ.0) GO TO 67
   IF (MA(I, J+1).EQ.0) GO TO 63
   IF (MA(I+1, J-1).EQ.0) GO TO 61
   IF (MA(I+1, J).EQ.0) GO TO 21
   M(I, J)=0
   GOTO 91
61 IF (MA(I+1, J).EQ.0) GO TO 21
   M(I, J)=0
   GOTO 91
63 IF (MA(I+1, J-1).EQ.0) GO TO 65
   IF (MA(I+1, J).EQ.0) GO TO 21
   M(I, J)=0
   GOTO 91
65 IF (MA(I+1, J).EQ.0) GO TO 21
   M(I, J)=0
   GOTO 91
67 IF (MA(I, J+1).EQ.0) GO TO 72
   IF (MA(I+1, J-1).EQ.0) GO TO 69
   IF (MA(I+1, J).EQ.0) GO TO 21
   M(I, J)=0
   GOTO 91

```



```

69  IF (MA(I+1,J).EQ.0) GO TO 70
    M(I,J)=0
    GOTOC91
70  IF (MA(I+1,J+1).EQ.0) GO TO 21
    M(I,J)=0
    GOTOC91
72  IF (MA(I+1,J-1).EQ.0) GO TO 21
    IF (MA(I+1,J).EQ.0) GO TO 21
    M(I,J)=0
    GOTOC91
1   M(I,J)=0
    GOTOC91
21  M(I,J)=1
91  CONTINUE
A1  CONTINUE
    PRINT170
    PRINT200
    PRINT120, ((LN(I,J),J=1,62),I=1,32)
170  FORMAT(1F1)
200  FORMAT(50X,'IMAGE LISSE',////)
120  FORMAT(4X,62T1)
    RETIR
    END

```

P R O G R A M M E S

C O N C E R N A N T

L A M E T H O D E

D E

R O S E N F E L D

E T

A L


```

DEFINE FILE 4=RD(R:256,G:1,RW,RN,FI)
DEFINE FILE 5=CD(R:256,G:1,RW,RN,FI)
INTEGER V(128),V1(128),V2(128),V3(128),V4(128)
INTEGER V5(128),V6(128),V7(128),V8(128),V9(128)
DIMENSION LVECT(128),IG(128),IT(128)
L=5000
I7=1
5 K=0
  IK=0
  DO 51 I=6308,6816,4
  READ(4,'I')(LVECT(J),J=1,128)
  J=0
  DO 52 LL=1,128
  J=J+1
52 V(LL)=LVECT(J)
  IF(IK.GT.1) GO TO 6
  K=K+1
  CALL INITIA(K,V,V1,V2,V3,V4,V5,V6,V7,V8)
  IF(I7.EQ.1) GO TO 1
  IF(I7.EQ.2) GO TO 2
  IF(I7.EQ.3) GO TO 3
  IF(K.LT.2) GO TO 51
  CALL FOCY1(V1,V2,IG,L)
  GO TO 9
  IF(I7.EQ.1) GO TO 11
  IF(I7.EQ.2) GO TO 22
  IF(I7.EQ.3) GO TO 33
  9 IK=IK+1
  DO 14 J=1,128
  V1(J)=V2(J)
  GO TO 51
  11 DO 15 J=1,128
  V2(J)=V(J)
  CALL FOCY1(V1,V2,IG,L)
  GO TO 9
  2 IF(K.LT.4) GO TO 51
  CALL FOCY2(V1,V2,V3,V4,IG,L)
  IK=IK+1
  19 DO 16 J=1,128
  V1(J)=V2(J)
  V2(J)=V3(J)
  V3(J)=V4(J)
  GO TO 51
  22 DO 17 J=1,128
  V4(J)=V(J)
  CALL FOCY2(V1,V2,V3,V4,IG,L)
  GO TO 19
  3 IF(K.LT.8) GO TO 51
  CALL FOCY3(V1,V2,V3,V4,V5,V6,V7,V8,IG,L)
  IK=IK+1
  29 DO 18 J=1,128
  V1(J)=V2(J)
  V2(J)=V3(J)
  V3(J)=V4(J)
  V4(J)=V5(J)
  V5(J)=V6(J)
  V6(J)=V7(J)
  V7(J)=V8(J)
  GO TO 51
  33 DO 20 J=1,128
  V8(J)=V(J)
  CALL FOCY3(V1,V2,V3,V4,V5,V6,V7,V8,IG,L)
  GO TO 29
  51 CONTINUE
  IF(I7.GT.1) GO TO 99
  DO 88 J=1,128
  88 IG(J)=0
  CALL RANG1(IG,L)
  CALL RANG1(IG,L)
  I7=I7+1
  GO TO 5
  99 IF(I7.EQ.3) GO TO 90
  DO 89 J=1,128
  89 IG(J)=0

```

```

CALL RANG1(IG,L)
CALL RANG1(IG,L)
CALL RANG1(IG,L)
CALL RANG1(IG,L)
CALL RANG1(IG,L)
J7=I7+1
GO TO 5
90 DO 91 J=1,128
91 IG(J)=0
CALL RANG1(IG,L)
CALL RANG1(IG,L)
CALL RANG1(IG,L)
STOP
END

```

```

SUBROUTINE INITIA(K,V,V1,V2,V3,V4,V5,V6,V7,VP)
INTEGER V(1)
INTEGER V1(1),V2(1),V3(1),V4(1)
INTEGER V5(1),V6(1),V7(1),V8(1)
IF(K.EQ.1) GO TO 10
IF(K.EQ.2) GO TO 11
IF(K.EQ.3) GO TO 12
IF(K.EQ.4) GO TO 13
IF(K.EQ.5) GO TO 14
IF(K.EQ.6) GO TO 15
IF(K.EQ.7) GO TO 16
IF(K.EQ.8) GO TO 17
10 DO 51 J=1,128
51 V1(J)=V(J)
GO TO 8
11 DO 52 J=1,128
52 V2(J)=V(J)
GO TO 8
12 DO 53 J=1,128
53 V3(J)=V(J)
GO TO 8
13 DO 54 J=1,128
54 V4(J)=V(J)
GO TO 8
14 DO 55 J=1,128
55 V5(J)=V(J)
GO TO 8
15 DO 56 J=1,128
56 V6(J)=V(J)
GO TO 8
16 DO 57 J=1,128
57 V7(J)=V(J)
GO TO 8
17 DO 58 J=1,128
58 V8(J)=V(J)
RETURN
END

```

```

SUBROUTINE TOY1(V1,V2,IG,L)
INTEGER V1(1),V2(1)
DIMENSION IG(1)
DO 10 J=2,128
M1=V1(J-1)+V1(J)+V2(J-1)+V2(J)
IG(J)=TRIX(M1/4)
10 CONTINUE
IG(1)=IG(2)
CALL RANG1(IG,L)
IF(L.GT.6002) GO TO 4
CALL RANG1(IG,L)
4 RETURN
END

```



```

SUBROUTINE NOY2(V1,V2,V3,V4,IG,L)
INTEGER V1(1),V2(1),V3(1),V4(1)
DIMENSION IG(1)
DO 10 J=3,127
M2=0
M2=M2+V1(J-2)+V1(J-1)+V1(J)+V1(J+1)
M2=M2+V2(J-2)+V2(J-1)+V2(J)+V2(J+1)
M2=M2+V3(J-2)+V3(J-1)+V3(J)+V3(J+1)
M2=M2+V4(J-2)+V4(J-1)+V4(J)+V4(J+1)
IG(J)=IFIX(M2/16)
CONTINUE
IG(1)=0
IG(2)=0
IG(128)=0
CALL RANG1(IG,L)
RETURN
END

```

10

```

SUBROUTINE NOY3(V1,V2,V3,V4,V5,V6,V7,V8,IG,L)
INTEGER V1(1),V2(1),V3(1),V4(1)
INTEGER V5(1),V6(1),V7(1),V8(1)
DIMENSION IG(1)
DO 10 J=5,125
M3=0
M3=M3+V1(J-4)+V1(J-3)+V1(J-2)+V1(J-1)
M3=M3+V1(J)+V1(J+1)+V1(J+2)+V1(J+3)
M3=M3+V2(J-4)+V2(J-3)+V2(J-2)+V2(J-1)
M3=M3+V2(J)+V2(J+1)+V2(J+2)+V2(J+3)
M3=M3+V3(J-4)+V3(J-3)+V3(J-2)+V3(J-1)
M3=M3+V3(J)+V3(J+1)+V3(J+2)+V3(J+3)
M3=M3+V4(J-4)+V4(J-3)+V4(J-2)+V4(J-1)
M3=M3+V4(J)+V4(J+1)+V4(J+2)+V4(J+3)
M3=M3+V5(J-4)+V5(J-3)+V5(J-2)+V5(J-1)
M3=M3+V5(J)+V5(J+1)+V5(J+2)+V5(J+3)
M3=M3+V6(J-4)+V6(J-3)+V6(J-2)+V6(J-1)
M3=M3+V6(J)+V6(J+1)+V6(J+2)+V6(J+3)
M3=M3+V7(J-4)+V7(J-3)+V7(J-2)+V7(J-1)
M3=M3+V7(J)+V7(J+1)+V7(J+2)+V7(J+3)
M3=M3+V8(J-4)+V8(J-3)+V8(J-2)+V8(J-1)
M3=M3+V8(J)+V8(J+1)+V8(J+2)+V8(J+3)
IG(J)=IFIX(M3/64)
CONTINUE
DO 11 J=1,4
IG(J)=0
DO 12 J=126,128
IG(J)=0
CALL RANG1(IG,L)
RETURN
END

```

10

11

12

```

SUBROUTINE RANG1(IG,L)
DIMENSION IG(1)
WRITE(5,L)(IG(J),J=1,128)
L=L+1
RETURN
END

```

```

DEFINE FILE 5=CD(R:256,G:1,RW,PN,FI)
INTEGER V(128),V1(128),V2(128),V3(128),V4(128),V5(128)
INTEGER V6(128),V7(128),V8(128),V9(128)
DIMENSION IG(128),LVECT(128),IT(128)
L=5384
K=0
IK=0
DO 51 I=5000,5127
READ(5,'I')(LVECT(J),J=1,128)
J=0
DO 52 LL=1,128
J=J+1
52 V(LL)=LVECT(J)
IF(IK.GE.1) GO TO 6
K=K+1
CALL CLASS1(K,V,V1,V2,V3)
IF(K.LT.3) GO TO 51
DO 10 J=2,127
10 IG(J)=IABS(V1(J+1)-V1(J-1))
IG(1)=0
IG(128)=0
CALL RANG1(IG,L)
DO 20 J=2,127
20 IT(J)=1
IT(1)=0
IT(128)=0
CALL RANG2(IT,L)
CALL DIF1(V1,V2,V3,IT,IG,L)
9 IK=IK+1
CALL SUIT1(V1,V2,V3)
GO TO 51
6 DO 14 J=1,128
14 V3(J)=V(J)
CALL DIF1(V1,V2,V3,IT,IG,L)
GO TO 9
51 CONTINUE
DO 60 J=2,127
60 IG(J)=IABS(V3(J+1)-V3(J-1))
IG(1)=0
IG(128)=0
DO 70 J=2,127
70 IT(J)=1
IT(1)=0
IT(128)=0
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL PRG2(V,V1,V2,V3,V4,V5,IG,IT,LVECT,L)
CALL PRG3(V,V1,V2,V3,V4,V5,V6,V7,V8,V9,IG,IT,LVECT,L)
CALL VERIF(LVECT,IG)
STOP
END

```

```

SUBROUTINE CLASS1(K,V1,V2,V3,V)
INTEGER V(1),V1(1),V2(1),V3(1)
IF(K.EQ.1) GO TO 10
IF(K.EQ.2) GO TO 12
IF(K.EQ.3) GO TO 14
10 DO51J=1,128
51 V1(J)=V(J)
GO TO 8
12 DO52J=1,128
52 V2(J)=V(J)
GO TO 8
14 DO53J=1,128
53 V3(J)=V(J)
8 RETURN
END

```



```

SUBROUTINE CTF1(V1,V2,V3,IG,IT,L)
INTEGER V1(1),V2(1),V3(1)
DIMENSION IG(1),IT(1)
DO 10 J=2,127
IH=IABS(V2(J+1)-V2(J-1))
ID1=IABS(V1(J+1)-V3(J-1))
IV=IABS(V1(J)-V3(J))
ID2=IABS(V1(J-1)-V3(J+1))
IG(J)=MAX0(IH,ID1,IV,ID2)
IF(IG(J).EQ.IH) GO TO 1
IF(IG(J).EQ.ID1) GO TO 2
IF(IG(J).EQ.IV) GO TO 3
IF(IG(J).EQ.ID2) GO TO 4
1 IT(J)=1
GO TO 10
2 IT(J)=2
GO TO 10
3 IT(J)=3
GO TO 10
4 IT(J)=4
10 CONTINUE
IG(1)=IABS(V1(1)-V3(1))
IG(128)=IABS(V1(128)-V3(128))
IT(1)=3
IT(128)=3
CALL RANG1(IG,L)
CALL RANG2(IT,L)
RETURN
END

```

```

SUBROUTINE SUIT1(V1,V2,V3)
INTEGER V1(1),V2(1),V3(1)
DO 30 J=1,128
V1(J)=V2(J)
V2(J)=V3(J)
RETURN
30 END

```

```

SUBROUTINE RANG1(IG,L)
DIMENSION IG(1)
WRITE(5,'L')(IG(J),J=1,128)
L=L+1
RETURN
END

```

```

SUBROUTINE RANG2(IT,L)
DIMENSION IT(1)
WRITE(5,'L')(IT(J),J=1,128)
L=L+1
RETURN
END

```

```

SUBROUTINE RANG3(IR,L)
DIMENSION IR(1)
WRITE(5,'L')(IR(J),J=1,128)
L=L+1
RETURN
END

```

```

SUBROUTINE SUIT2(V1,V2,V3,V4,V5)
INTEGER V1(1),V2(1),V3(1),V4(1),V5(1)
DO 30 J=1,128
V1(J)=V2(J)
V2(J)=V3(J)
V3(J)=V4(J)
V4(J)=V5(J)
RETURN
30 END

```

```

SUBROUTINE PRGG2(V,V1,V2,V3,V4,V5,IG,IT,LVFCT,L)
INTEGER V(1),V2(1),V3(1),V4(1),V5(1)
DIMENSION IG(1),IT(1),LVFCT(1)
K=0
IK=0
DO 10 J=1,128
IG(J)=0
10 IT(J)=0
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
DO 51 I=130,254
READ(5,I)(LVFCT(J),J=1,128)
J=0
DO 52 LL=1,128
J=J+1
52 V(LL)=LVFCT(J)
IF(IK.GE.1) GO TO 6
K=K+1
CALL CLASS2(K,V,V1,V2,V3,V4,V5)
IF(K.LT.5) GO TO 51
CALL DIF2(V1,V2,V3,V4,V5,IG,IT,L)
9 IK=IK+1
CALL SUT2(V1,V2,V3,V4,V5)
GO TO 51
6 DO 14 J=1,128
14 VS(J)=V(J)
CALL DIF2(V1,V2,V3,V4,V5,IG,IT,L)
GO TO 9
51 CONTINUE
DO 60 J=1,128
IG(J)=0
60 IT(J)=0
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
RETURN
END

```

```

SUBROUTINE CLASS2(K,V,V1,V2,V3,V4,V5)
INTEGER V1(1),V2(1),V3(1),V4(1),V5(1)
INTEGER V(1)
IF(K.EQ.1) GO TO 10
IF(K.EQ.2) GO TO 12
IF(K.EQ.3) GO TO 14
IF(K.EQ.4) GO TO 16
IF(K.EQ.5) GO TO 18
10 DO 51 J=1,128
51 V1(J)=V(J)
GO TO 8
12 DO 52 J=1,128
52 V2(J)=V(J)
GO TO 8
14 DO 53 J=1,128
53 V3(J)=V(J)
GO TO 8
16 DO 54 J=1,128
54 V4(J)=V(J)
GO TO 8
18 DO 55 J=1,128
55 V5(J)=V(J)
RETURN
END

```



```

SUBROUTINE DIF2(V1,V2,V3,V4,V5,IG,IT,L)
INTEGER V1(1),V2(1),V3(1),V4(1),V5(1)
DIMENSION IG(1),IT(1)
DO10J=5,125
IH=IABS(V3(J+2)-V3(J-2))
ID1=IABS(V1(J+2)-V5(J-2))
IV=IABS(V1(J)-V5(J))
ID2=IABS(V1(J-2)-V5(J+2))
IG(J)=MAX0(IH,ID1,IV,ID2)
IF(IG(J).EQ.IH) GO TO 1
IF(IG(J).EQ.ID1) GO TO 2
IF(IG(J).EQ.IV) GOTOC3
IF(IG(J).EQ.ID2) GO TO 4
1 IT(J)=1
GO TO 10
2 IT(J)=2
GO TO 10
3 IT(J)=3
GO TO 10
4 IT(J)=4
GO TO 10
10 CONTINUE
DO40J=1,4
IG(J)=0
40 IT(J)=0
DO 50 J=126,128
IG(J)=0
50 IT(J)=0
CALL RANG1(IG,L)
CALL RANG2(IT,L)
RETURN
END

```

```

SUBROUTINE DIF3(V1,V2,V3,V4,V5,V6,V7,V8,V9,IG,IT,L)
INTEGER V1(1),V2(1),V3(1),V4(1),V5(1)
INTEGER V6(1),V7(1),V8(1),V9(1)
DIMENSION IG(1),IT(1)
DO10J=2,127
IH=IABS(V5(J+1)-V5(J-1))
ID1=IABS(V1(J+1)-V9(J-1))
IV=IABS(V1(J)-V9(J))
ID2=IABS(V1(J-1)-V9(J+1))
IG(J)=MAX0(IH,ID1,IV,ID2)
IF(IG(J).EQ.IH) GOTOC1
IF(IG(J).EQ.ID1) GOTOC2
IF(IG(J).EQ.IV) GOTOC3
IF(IG(J).EQ.ID2) GOTOC4
1 IT(J)=1
GO TO 10
2 IT(J)=2
GO TO 10
3 IT(J)=3
GO TO 10
4 IT(J)=4
GO TO 10
10 CONTINUE
DO 20 J=1,8
IG(J)=0
20 IT(J)=0
DO 30 J=122,128
IG(J)=0
30 IT(J)=0
CALL RANG1(IG,L)
CALL RANG2(IT,L)
RETURN
END

```

```

SUBROUTINE CLASS3(K,V,V1,V2,V3,V4,V5,V6,V7,VA,VA9)
INTEGER V1(1),V2(1),V3(1),V4(1),V5(1)
INTEGER V6(1),V7(1),V8(1),V9(1)
INTEGER V(1)
IF(K.EQ.1) GO TO 10
IF(K.EQ.2) GO TO 12
IF(K.EQ.3) GO TO 14
IF(K.EQ.4) GOTO 16
IF(K.EQ.5) GOTO 18
IF(K.EQ.6) GOTO 20
IF(K.EQ.7) GOTO 22
IF(K.EQ.8) GOTO 24
IF(K.EQ.9) GOTO 26
10 DO 51 J=1,12P
51 V1(J)=V(J)
GO TO P
12 DO 52 J=1,12P
52 V2(J)=V(J)
GO TO P
14 DO 53 J=1,12P
53 V3(J)=V(J)
GO TO P
16 DO 54 J=1,12P
54 V4(J)=V(J)
GO TO P
18 DO 55 J=1,12P
55 V5(J)=V(J)
GO TO P
20 DO 56 J=1,12P
56 V6(J)=V(J)
GO TO P
22 DO 57 J=1,12P
57 V7(J)=V(J)
GO TO P
24 DO 58 J=1,12P
58 VA(J)=V(J)
GO TO P
26 DO 59 J=1,12P
59 V8(J)=V(J)
RETURN
END

```

```

SUBROUTINE SUIT3(V1,V2,V3,V4,V5,V6,V7,VA,VA9)
INTEGER V1(1),V2(1),V3(1),V4(1),V5(1)
INTEGER V6(1),V7(1),V8(1),V9(1)
DO 1P J=1,12P
1P V1(J)=V2(J)
V2(J)=V3(J)
V3(J)=V4(J)
V4(J)=V5(J)
V5(J)=V6(J)
V6(J)=V7(J)
V7(J)=VA(J)
RETURN
END

```

```

SUBROUTINE PROG3(V,V1,V2,V3,V4,V5,V6,V7,V8,V9,LVECT,IG,IT,L)
INTEGER V1(1),V2(1),V3(1),V4(1),V5(1)
INTEGER V6(1),V7(1),V8(1),V9(1)
INTEGER V(1)
DIMENSION IG(1),IT(1),LVECT(1)
K=0
IK=0
DO10J=1,128
IG(J)=0
IT(J)=0
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
DO 51 I=5260,5380
READ(S'I)(LVECT(J),J=1,128)
J=0
DO52LL=1,128
J=J+1
V(LL)=LVECT(J)
IF(IK.GE.1)GOTO6
K=K+1
CALL CLASS3(K,V,V1,V2,V3,V4,V5,V6,V7,V8,V9)
IF(K.LT.9)GOTO51
CALL DIF3(V1,V2,V3,V4,V5,V6,V7,V8,V9,IG,IT,L)
IK=IK+1
CALL SUITE3(V1,V2,V3,V4,V5,V6,V7,V8,V9)
GOTO51
DO14J=1,128
V9(J)=V(J)
CALL DIF3(V1,V2,V3,V4,V5,V6,V7,V8,V9,IG,IT,L)
GO TO 9
CONTINUE
DO60J=1,128
IG(J)=0
IT(J)=0
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG1(IG,L)
CALL RANG2(IT,L)
RETURN
END

```

10

52

9

6

14

51

60

3^{ème}PARTIE DU PROGRAMME DE LA METHODE DE ROSENFELD ET AL

```

DEFINE FILE 5=CD(P:256,G:1,RW,BN,FI)
DIMENSION LV1(128),LV2(128),LV3(128)
DIMENSION LT1(128),LT2(128),LT3(128),IR(128)
DIMENSION IG(128),IT(128)
L=6200
DO 51 I=5384,5638,2
M1=I+1
M2=I+256
M3=M1+1
K=M1+1
M3=K+1
READ(5'I')(LV1(J),J=1,128)
READ(5'L')(LV2(J),J=1,128)
READ(5'K')(LV3(J),J=1,128)
READ(5'M1')(LT1(J),J=1,128)
READ(5'M2')(LT2(J),J=1,128)
READ(5'M3')(LT3(J),J=1,128)
DO 40 J=1,128
IC=3*LV2(J)
IG=4*LV1(J)
IF=3*LV3(J)
IF=4*LV2(J)
IF(IC.GE.ID) GO TO 1
IF(LV3(J).EQ.0) GO TO 2
IF(IE.GE.IH) GO TO 3
IG(J)=LV1(J)
IT(J)=LT1(J)
IR(J)=1
GO TO 40
3 IG(J)=LV3(J)
IT(J)=LT3(J)
IR(J)=3
GO TO 40
2 IG(J)=LV1(J)
IT(J)=LT1(J)
IR(J)=1
GO TO 40
1 IF(IE.GE.IH) GO TO 4
IG(J)=LV2(J)
IT(J)=LT2(J)
IR(J)=2
GO TO 40
4 IG(J)=LV3(J)
IT(J)=LT3(J)
IR(J)=3
40 CONTINUE
CALL RANG1(IG,L)
CALL RANG2(IT,L)
CALL RANG3(IR,L)
51 CONTINUE
STOP
END

```

```

SUBROUTINE RANG1(IG,L)
DIMENSION IG(1)
WRITE(5'L')(IG(J),J=1,128)
L=L+1
RETURN
END

```

```

SUBROUTINE RANG2(IT,L)
DIMENSION IT(1)
WRITE(5'L')(IT(J),J=1,128)
L=L+1
RETURN
END

```

```

SUBROUTINE RANG3(IR,L)
DIMENSION IR(1)
WRITE(5'L')(IR(J),J=1,128)
L=L+1
RETURN
END

```

PARTIE DU PROGRAMME DE LA METHODE DE ROSENFIELD ET AL

```

DEFINE FILE 5=CD(R:256,G:1,RW,BN,FT)
DIMENSION M1(128),M2(128),M3(128),M4(128),M5(128),M6(128),M7(128)
DIMENSION M8(128),M9(128),M(128),LV1(128),LV2(128),LV3(128)
DIMENSION ID2(128),ID3(128),ID4(128),ID5(128)
DIMENSION ID6(128),ID7(128),ID8(128),ID9(128),IP(128)
COMMON/TAA/ID1(128),IT(128),ID(128)
COMMON/TRA/IT1(128),IT2(128),IT3(128)
COMMON/TCA/IT4(128),IT6(128),IT7(128)
COMMON/TCA/IT8(128),IT9(128)
COMMON/TEA/IT5(128)
DIMENSION IK(128),IL(128),IM(128)
DATA BLANC,IPPOINT/1H,1H./
N=0
IK=0
DO 51 I=2200,2581,3
L=I+1
K=L+1
READ(5,'I')(LV1(J),J=1,128)
READ(5,'L')(LV2(J),J=1,128)
READ(5,'K')(LV3(J),J=1,128)
DO 52 J=1,128
M(J)=LV1(J)
ID(J)=LV2(J)
IT(J)=LV3(J)
IF(IK.FE.1) GO TO 13
N=N+1
CALL ORDRF(M,M1,M2,M3,M4,M5,M6,M7,M8,M9,ID2,ID3,ID4,ID5,ID6,ID7,
*ID8,ID9,N)
IF(N.LT.9)GOTO 51
CALL TRACE1(M1,M2,M3,ID2,IP)
CALL TRACE2(M1,M2,M3,M4,M5,ID3,IP)
CALL TRACE3(IM,M1,M2,M3,M4,M5,M6,IM,IL,IP,ID3,ID4)
CALL TRACE(M1,M2,M3,M4,M5,M6,M7,M8,M9,ID5,IP)
IK=IK+1
CALL SUIT(M1,M2,M3,M4,M5,M6,M7,M8,M9,ID2,ID3,ID4,ID5,ID6,ID7,ID8
*,ID9)
GOTO 51
DO 43 J=1,128
M9(J)=M(J)
ID9(J)=ID(J)
IT9(J)=IT(J)
CALL TRACE(M1,M2,M3,M4,M5,M6,M7,M8,M9,ID5,IP)
GOTO 9
51 CONTINUE
CALL SUIT(M1,M2,M3,M4,M5,M6,M7,M8,M9,ID2,ID3,ID4,ID5,ID6,ID7,ID8,
*ID9)
CALL TRACE(M1,M2,M3,M4,M5,M6,M7,M8,M9,ID5,IP)
CALL SUIT(M1,M2,M3,M4,M5,M6,M7,M8,M9,ID2,ID3,ID4,ID5,ID6,ID7,ID8,
*ID9)
CALL TRACE(M1,M2,M3,M4,M5,M6,M7,M8,M9,ID5,IP)
CALL SUIT(M1,M2,M3,M4,M5,M6,M7,M8,M9,ID2,ID3,ID4,ID5,ID6,ID7,ID8,
*ID9)
CALL TRACE(M1,M2,M3,M4,M5,M6,M7,M8,M9,ID5,IP)
STOP
END

```

```

SUBROUTINE CRDRE(M,M1,M2,M3,M4,M5,M6,M7,M8,M9, ID2, ID3, ID4, ID5, ID6
*, ID7, ID8, ID9, N)
  DIMENSION M(1), M2(1), M3(1), M4(1), M5(1), M6(1), M7(1), M8(1), M9(1)
  DIMENSION ID2(1), ID3(1), ID4(1), ID5(1), ID6(1), ID7(1), ID8(1), ID9(1)
  DIMENSION M1(1)
  COMMON/TAA/ID1(128), IT(128), TD(128)
  COMMON/TRA/IT1(128), IT2(128), IT3(128)
  COMMON/TCA/IT4(128), IT6(128), IT7(128)
  COMMON/TDA/IT8(128), IT9(128)
  COMMON/TFA/ITS(128)
  IF (N.EQ.1) GOTO 1
  IF (N.EQ.2) GOTO 2
  IF (N.EQ.3) GOTO 3
  IF (N.EQ.4) GOTO 4
  IF (N.EQ.5) GOTO 5
  IF (N.EQ.6) GOTO 6
  IF (N.EQ.7) GOTO 7
  IF (N.EQ.8) GOTO 8
  IF (N.EQ.9) GOTO 9
1  DO 10 J=1, 128
   M1(J)=M(J)
   ID1(J)=ID(J)
   IT1(J)=IT(J)
10  GO TO 100
2  DO 20 J=1, 128
   M2(J)=M(J)
   ID2(J)=ID(J)
   IT2(J)=IT(J)
20  GO TO 100
3  DO 30 J=1, 128
   M3(J)=M(J)
   ID3(J)=ID(J)
   IT3(J)=IT(J)
30  GO TO 100
4  DO 40 J=1, 128
   M4(J)=M(J)
   ID4(J)=ID(J)
   IT4(J)=IT(J)
40  GO TO 100
5  DO 50 J=1, 128
   M5(J)=M(J)
   ID5(J)=ID(J)
   IT5(J)=IT(J)
50  GO TO 100
6  DO 60 J=1, 128
   M6(J)=M(J)
   ID6(J)=ID(J)
   IT6(J)=IT(J)
60  GO TO 100
7  DO 70 J=1, 128
   M7(J)=M(J)
   ID7(J)=ID(J)
   IT7(J)=IT(J)
70  GO TO 100
8  DO 80 J=1, 128
   M8(J)=M(J)
   ID8(J)=ID(J)
   IT8(J)=IT(J)
80  GO TO 100
9  DO 90 J=1, 128
   M9(J)=M(J)
   ID9(J)=ID(J)
   IT9(J)=IT(J)
90  RETURN
100 END

```



```

SUBROUTINE TRACE1(M1,M2,M3,IP,IP)
DIMENSION M1(1),M2(1),M3(1),IP(1)
DATA IBLANC, IPOINT/10, 10./
DO10 J=2, 127
IF (IP2(J).EQ.1) GO TO 1
IF (IP2(J).EQ.2) GO TO 2
IF (IP2(J).EQ.3) GO TO 3
IF (M1(J+1).EQ.M2(J)) GO TO 99
IF (M3(J-1).EQ.M2(J)) GO TO 99
GO TO 99
3 IF (M2(J+1).EQ.M2(J)) GO TO 99
IF (M2(J-1).EQ.M2(J)) GO TO 99
GO TO 99
2 IF (M1(J-1).EQ.M2(J)) GO TO 99
IF (M3(J+1).EQ.M2(J)) GO TO 99
GO TO 99
1 IF (M1(J).EQ.M2(J)) GO TO 99
IF (M3(J).EQ.M2(J)) GO TO 99
99 IP(J)=IPOINT
GO TO 10
99 IP(J)=IBLANC
10 CONTINUE
PRINT 140, (IP(J), J=1, 128)
FORMAT (2X, 12PA1)
140 RETURN
END

```

```

SUBROUTINE TRACE2(M1,M2,M3,M4,M5,IP,IP)
DIMENSION M1(1),M2(1),M3(1),M4(1),M5(1),IP(1)
DATA IBLANC, IPOINT/10, 10./
DO20 J=2, 127
IF (IP3(J).EQ.1) GO TO 4
IF (IP3(J).EQ.2) GO TO 5
IF (IP3(J).EQ.3) GO TO 6
IF (IT3(J).EQ.1) GO TO 7
IF (M1(J+2).EQ.M3(J)) GO TO 99
IF (M5(J-2).EQ.M3(J)) GO TO 99
GO TO 99
7 IF (M2(J+1).EQ.M3(J)) GO TO 99
IF (M4(J-1).EQ.M3(J)) GO TO 99
GO TO 99
6 IF (IT3(J).EQ.1) GO TO 8
IF (M3(J+2).EQ.M3(J)) GO TO 99
IF (M3(J-2).EQ.M3(J)) GO TO 99
GO TO 99
8 IF (M3(J-1).EQ.M3(J)) GO TO 99
IF (M3(J+1).EQ.M3(J)) GO TO 99
GO TO 99
5 IF (IT3(J).EQ.1) GO TO 9
IF (M1(J-2).EQ.M3(J)) GO TO 99
IF (M5(J+2).EQ.M3(J)) GO TO 99
GO TO 99
9 IF (M2(J-1).EQ.M3(J)) GO TO 99
IF (M4(J+1).EQ.M3(J)) GO TO 99
GO TO 99
4 IF (IT3(J).EQ.1) GO TO 10
IF (M1(J).EQ.M3(J)) GO TO 99
IF (M5(J).EQ.M3(J)) GO TO 99
GO TO 99
10 IF (M2(J).EQ.M3(J)) GO TO 99
IF (M4(J).EQ.M3(J)) GO TO 99
99 IP(J)=IPOINT
GO TO 20
99 IP(J)=IBLANC
20 CONTINUE
PRINT 140, (IP(J), J=1, 128)
FORMAT (2X, 12PA1)
140 RETURN
END

```

```

SUBROUTINE TRACE3(M1,M2,M3,M4,M5,M6,IN,IL,IP,IO3,IO4)
DIMENSION IT(1),M1(1),M2(1),M3(1),M4(1),M5(1),M6(1),IN(1)
DIMENSION IL(1),IP(1),IO3(1),IO4(1)
COMMON/TAA/ID1(128),IT(128),ID(128)
COMMON/TFA/IT1(128),IT2(128),IT3(128)
COMMON/TCA/IT4(128),IT6(128),IT7(128)
COMMON/TDA/IT8(128),IT9(128)
DATA IPLAN,IPCINT/1H,1F./
DO 30 J=1,128
  IN(J)=M1(J)
  M1(J)=M2(J)
  M2(J)=M3(J)
  M3(J)=M4(J)
  M4(J)=M5(J)
  M5(J)=M6(J)
  IN(J)=IT3(J)
  IT3(J)=IT4(J)
  IL(J)=IO3(J)
  IO3(J)=IO4(J)
  CALL TRACE2(M1,M2,M3,M4,M5,IO3,IP)
DO 40 J=1,128
  M6(J)=M5(J)
  M5(J)=M4(J)
  M4(J)=M3(J)
  M3(J)=M2(J)
  M2(J)=M1(J)
  M1(J)=IN(J)
  IT4(J)=IT3(J)
  IT3(J)=IN(J)
  IO4(J)=IO3(J)
  IO3(J)=IL(J)
CONTINUE
RETURN
END

```

```

SUBROUTINE TRACE(M1,M2,M3,M4,M5,M6,M7,M8,M9,IDS,IP)
DIMENSION I(1),M2(1),M3(1),M4(1),M5(1),M6(1),M7(1),M9(1)
DIMENSION M8(1),IDS(1),IP(1)
COMMON/TEA/ITS(128)
DATA IBLANK,IPCNT/1H ,1F./
DO 20 J=2,127
IF (IDS(J).EQ.1)GOTO1
IF (IDS(J).EQ.2)GOTO2
IF (IDS(J).EQ.3)GOTO3
IF (ITS(J).EQ.1)GOTO4
IF (ITS(J).EQ.2)GOTO5
IF (M1(J+4).EQ.M5(J))GOTO98
IF (M1(J-4).EQ.M5(J))GOTO98
GO TO 99
E
IF (M7(J+2).EQ.M5(J))GOTO98
IF (M7(J-2).EQ.M5(J))GOTO98
GOTO99
4
IF (M4(J+1).EQ.M5(J))GOTO98
IF (M6(J+1).EQ.M5(J))GOTO98
GOTO99
3
IF (ITS(J).EQ.1)GOTO6
IF (ITS(J).EQ.2)GOTO7
IF (M5(J+4).EQ.M5(J))GOTO98
IF (M5(J-4).EQ.M5(J))GOTO98
GOTO99
7
IF (M5(J+2).EQ.M5(J))GOTO98
IF (M5(J-2).EQ.M5(J))GOTO98
GOTO99
6
IF (M5(J+1).EQ.M5(J))GOTO98
IF (M5(J-1).EQ.M5(J))GOTO98
GOTO99
2
IF (ITS(J).EQ.1)GOTO8
IF (ITS(J).EQ.2)GOTO9
IF (M1(J-8).EQ.M5(J))GOTO98
IF (M9(J+4).EQ.M5(J))GOTO98
GOTO99
9
IF (M3(J-2).EQ.M5(J))GOTO98
IF (M7(J+2).EQ.M5(J))GOTO98
GOTO99
8
IF (M4(J-1).EQ.M5(J))GOTO98
IF (M6(J+1).EQ.M5(J))GOTO98
GOTO99
1
IF (ITS(J).EQ.1)GOTO10
IF (ITS(J).EQ.2)GOTO11
IF (M1(J).EQ.M5(J))GOTO98
IF (M9(J).EQ.M5(J))GOTO98
GOTO99
11
IF (M3(J).EQ.M5(J)) GO TO 98
IF (M7(J).EQ.M5(J))GOTO98
GOTO99
10
IF (M4(J).EQ.M5(J))GOTO98
IF (M6(J).EQ.M5(J))GOTO98
98
IP(J)=IPCNT
GOTO90
90
IF (J)=IBLANK
CONTINUE
PRINT140,(IP(J),J=1,128)
140
FORMAT(2X,12A1)
RETURN
END

```



```

SUBROUTINE SUIT(M1,M2,M3,M4,M5,M6,M7,M8,M9,IC ,ID3,ID4,ID5,ID6,
*ID7,ID8 ,ID9)
COMMON/TAA/ID1(128),IT(128),ID(128)
COMMON/TRA/IT1(128),IT2(128),IT3(128)
COMMON/TCA/IT4(128),IT6(128),IT7(128)
COMMON/TDA/IT8(128),IT9(128)
COMMON/TEA/IT5(128)
DIMENSION M1(1),M2(1),M3(1),M4(1),M5(1),M6(1),M7(1),M8(1),M9(1)
DIMENSION ID2(1),ID3(1),ID4(1),ID5(1),ID6(1),ID7(1),ID8(1),ID9(1)
DO50J=1,128
M1(J)=M2(J)
M2(J)=M3(J)
M3(J)=M4(J)
M4(J)=M5(J)
M5(J)=M6(J)
M6(J)=M7(J)
M7(J)=M8(J)
M8(J)=M9(J)
50 DO60J=1,128
IT1(J)=IT2(J)
IT2(J)=IT3(J)
IT3(J)=IT4(J)
IT4(J)=IT5(J)
IT5(J)=IT6(J)
IT6(J)=IT7(J)
IT7(J)=IT8(J)
IT8(J)=IT9(J)
60 DO70J=1,128
ID1(J)=ID2(J)
ID2(J)=ID3(J)
ID3(J)=ID4(J)
ID4(J)=ID5(J)
ID5(J)=ID6(J)
ID6(J)=ID7(J)
ID7(J)=ID8(J)
ID8(J)=ID9(J)
70 RETURN
END

```

P R O G R A M M E S

C O N C E R N A N T

L E S

D I F F E R E N T E S

T R A N S F O R M A T I O N S

L O C A L E S

```

C      PROGRAMME PRINCIPAL VALABLE
C      POUR LES TRANSFORMATIONS LOCALES
C      MOYENNE, ECART A LA MOYENNE, CONTRASTE, ET ECART TYPE
      DEFINE FILE 4=BD(R:256,G:1,RW,BN,FI)
      DEFINE FILE 5=CD(R:256,G:1,RW,BN,FI)
      DIMENSION LVECT(128),IM(128),IG(64)
      INTEGER V(128),V1(128),V2(128),V3(128)
      L=4978
      K=0
      IZ=0
      DO 10 I=6365,6873,8
      READ(4,I)(LVECT(J),J=1,128)
      DO 11 J=1,128
11      V(J)=LVECT(J)
      IF (IZ.GE.1) GO TO 12
      K=K+1
      CALL INITIA(K,V,V1,V2,V3)
      IF (K.LT.3) GO TO 10
      CALL MOYENN(V1,V2,V3,IM,IG,L)
9      IZ=IZ+1
      CALL SUITE(V1,V2,V3)
      GO TO 10
12      DO 13 J=1,128
13      V3(J)=V(J)
      CALL MOYENN(V1,V2,V3,IM,IG,L)
      GO TO 9
10      CONTINUE
      STOP
      END

      SUBROUTINE INITIA(K,V,V1,V2,V3)
      INTEGER V(1),V1(1),V2(1),V3(1)
      IF (K.EQ.1) GO TO 1
      IF (K.EQ.2) GO TO 2
      IF (K.EQ.3) GO TO 3
1      DO 11 J=1,128
11      V1(J)=V(J)
      GO TO 8
2      DO 22 J=1,128
22      V2(J)=V(J)
      GO TO 8
3      DO 33 J=1,128
33      V3(J)=V(J)
8      RETURN
      END

      SUBROUTINE SUITE(V1,V2,V3)
      INTEGER V1(1),V2(1),V3(1)
      DO 66 J=1,128
66      V1(J)=V2(J)
      V2(J)=V3(J)
      RETURN
      END

```



```

C      SUBROUTINE MOYENN(V1,V2,V3,IM,IG,L)
      S/P CALCULANT LA MOYENNE
      INTEGER V1(1),V2(1),V3(1)
      DIMENSION IM(1),IG(1)
      DO 44 J=2,127
      M=0
      M=M+V1(J-1)+V1(J)+V1(J+1)
      M=M+V2(J-1)+V2(J)+V2(J+1)
      M=M+V3(J-1)+V3(J)+V3(J+1)
44     IM(J)=IFIX(M/9)
      WRITE(5,L)(IM(J),J=1,128)
      L=L+1
      RETURN
      END

```

```

C      SUBROUTINE MOYENN(V1,V2,V3,IM,IG,L)
      S/P CALCULANT L'ECART TYPE
      INTEGER V1(1),V2(1),V3(1)
      DIMENSION IM(1),IG(1)
      DO 44 J=2,127
      M1=0
      M1=M1+V1(J-1)**2+V1(J)**2+V1(J+1)**2
      M1=M1+V2(J-1)**2+V2(J)**2+V2(J+1)**2
      M1=M1+V3(J-1)**2+V3(J)**2+V3(J+1)**2
      M1=IFIX(M1/9)
      M2=0
      M2=M2+V1(J-1)+V1(J)+V1(J+1)
      M2=M2+V2(J-1)+V2(J)+V2(J+1)
      M2=M2+V3(J-1)+V3(J)+V3(J+1)
      M2=IFIX(M2/9)**2
44     IM(J)=SGRT(M1-M2)
      WRITE(5,L)(IM(J),J=1,128)
      L=L+1
      RETURN
      END

```

```

C      SUBROUTINE MOYENN(V1,V2,V3,IM,IG,L)
      S/P CALCULANT LE CONTRASTE
      INTEGER V1(1),V2(1),V3(1)
      DIMENSION IM(1),IG(1)
      DO 44 J=2,127
      M=0
      M=M+V1(J-1)+V1(J)+V1(J+1)
      M=M+V2(J-1)+V2(J)+V2(J+1)
      M=M+V3(J-1)+V3(J)+V3(J+1)
      N=IFIX(M/9)
      IM(J)=V2(J)*2-N
      IM(J)=IM(J)+5
44     CONTINUE
      WRITE(5,L)(IM(J),J=1,128)
      L=L+1
      RETURN
      END

```

```

      SUBROUTINE MOYENN(V1,V2,V3,IM,IG,L)
      INTEGER V1(1),V2(1),V3(1)
      DIMENSION IM(1),IG(1)
      DO 44 J=2,127
      M=0
      M=M+V1(J-1)+V1(J)+V1(J+1)
      M=M+V2(J-1)+V2(J)+V2(J+1)
      M=M+V3(J-1)+V3(J)+V3(J+1)
      N=IFIX(M/9)
      IM(J)=V2(J)-N
      IM(J)=IM(J)+20
44     CONTINUE
      WRITE(5,L)(IM(J),J=1,128)
      L=L+1
      RETURN
      END

```

```

DEFINE FILE 4=RD(R:256,G:1,RW,RN,FI)
DEFINE FILE 5=CD(R:256,G:1,RW,RN,FI)
INTEGER V1,V2,V3,V4,V5
DIMENSION V(128),V1(128),V2(128),V3(128)
DIMENSION V4(128),V5(128),LVECT(128)
N=5104
K=0
IZ=0
DO 51 I=6365,6873,4
READ(4,'I')(LVECT(J),J=1,128)
J=0

DO 52 LL=1,128
J=J+1
52 V(LL)=LVECT(J)
IF(IZ.GE.1) GO TO 6
K=K+1
CALL INITIA(K,V1,V2,V3,V4,V)
IF(K.LT.4) GO TO 51
9 CALL GRADA(V1,V2,V3,V4,V5,N)
IZ=IZ+1
CALL ISUIT(V1,V2,V3,V4)
GO TO 51
6 DO 14 J=1,128
14 V4(J)=V(J)
CALL GRADA(V1,V2,V3,V4,V5,N)
GO TO 9
GO TO 9
51 CONTINUE
CALL RESTE(V2,V3,V4,V5,N)
STOP
END

SUBROUTINE INITIA(K,V1,V2,V3,V4,V)
INTEGER V1(1),V2(1),V3(1),V4(1),V(1)
IF(K.EQ.1) GO TO 10
IF(K.EQ.2) GO TO 12
IF(K.EQ.3) GO TO 14
IF(K.EQ.4) GO TO 16
10 DO 51 J=1,128
51 V1(J)=V(J)
GO TO 8
12 DO 52 J=1,128
52 V2(J)=V(J)
GO TO 8
14 DO 53 J=1,128
53 V3(J)=V(J)
GO TO 8
16 DO 54 J=1,128
54 V4(J)=V(J)
8 RETURN
END

SUBROUTINE RESTE(V2,V3,V4,V5,N)
INTEGER V2(1),V3(1),V4(1),V5(1)
DO 80 L=1,127
80 V5(L)=IABS(V2(L)-V3(L+1))+IABS(V3(L)-V2(L+1))
CONTINUE
V5(L)=V5(L-1)
WRITE(5,'N')(V5(J),J=1,128)
N=N+1
DO 85 L=1,127
85 V5(L)=IABS(V3(L)-V4(L+1))+IABS(V4(L)-V3(L+1))
CONTINUE
V5(L)=V5(L-1)
WRITE(5,'N')(V5(J),J=1,128)
N=N+1
WRITE(5,'N')(V5(J),J=1,128)
RETURN
END

```

```

SUBROUTINE GRADA(V1,V2,V3,V4,V5,N)
INTEGER V1(1),V2(1),V3(1),V4(1),V5(1)
DO 40 L=1,128
IC=IABS(V2(L+1)-V3(L+2))+IABS(V3(L+1)-V2(L+2))
ID=IABS(V1(L)-V4(L+3))+IABS(V4(L)-V1(L+3))
IW=IABS(V1(L+1)-V4(L+2))+IABS(V3(L)-V2(L+3))
IZ=IABS(V1(L+2)-V4(L+1))+IABS(V2(L)-V3(L+3))
IM1=SQRT(IC*ID)
IM2=SQRT(IW*IZ)
IM3=SQRT(IM1)
IM4=SQRT(IM2)
V5(L)=IM3*IM4
40 CONTINUE
DO 70 L=126,127
V5(L)=IABS(V1(L)-V2(L+1))+IABS(V2(L)-V1(L+1))
70 CONTINUE
V5(L)=V5(L-1)
WRITE(5,N)(V5(J),J=1,128)
N=N+1
RETURN
END

```

```

SUBROUTINE ISUIT(V1,V2,V3,V4)
INTEGER V1(1),V2(1),V3(1),V4(1)
DO 41 J=1,128
V1(J)=V2(J)
V2(J)=V3(J)
V3(J)=V4(J)
41 RETURN
END

```

```

SUBROUTINE MOYENN(V1,V2,V3,IM,IG,L)
S/P CALCULANT LE CONTRASTE DU GRADIENT
DIMENSIONS DE LA FENETRE SONT (3*3)
INTEGER V1(1),V2(1),V3(1)
DIMENSION IM(1),IG(1)
DO 44 J=2,127
M=0
M=M+V1(J-1)+V1(J)+V1(J+1)
M=M+V2(J-1)+V2(J)+V2(J+1)
M=M+V3(J-1)+V3(J)+V3(J+1)
N=IFIX(M/9)
IM(J)=V2(J)*2-N
IG(J)=IM(J)+15
44 CONTINUE
WRITE(5,L)(IM(J),J=1,128)
L=L+1
RETURN
END

```



```

DEFINE FILE 5=CD(R:256,G:1,RK,RN,FI)
DIMENSION LV(128),IFREQ(300)
DO 50 I=1,300
IFREQ(I)=0
MIN=256
MAX=0
DO 60 I=5232,5359
READ(5,I)(LV(J),J=1,128)
DO 70 K=1,128
IA=LV(K)
MIN=MIN0(MIN,IA)
MAX=MAX0(MAX,IA)
KI=IA+1
IFREQ(KI)=IFREQ(KI)+1
70 CONTINUE
60 PRINT160,(MIN,MAX)
160 FORMAT(10X,'MIN FGAL A',I3,3X,'MAX FGAL A',I3)
CALL HISTO(MIN,MAX,IFREQ)
STOP
END

SUBROUTINE HISTO(MIN,MAX,IFREQ)
DIMENSION IFREQ(1)
DIMENSION LIG(120)
DATA IH,I2,IMOINS,II,IRLANC/1HH,1HZ,1H-,1HI,1H /
IMAX=IFREQ(1)
DO 60 J=1,256
IF(IMAX.GT. IFREQ(J)) GO TO 60
IMAX=IFREQ(J)
60 CONTINUE
IF(IMAX.LE.80)GOTO2
K=IMAX/80
GO TO 3
2 K=1
3 PRINT 154
PRINT155
PRINT205
PRINT206
PRINT 151,K
PRINT 155
PRINT 156
PRINT 153
KI=MIN+1
ML=MAX+1
DO 62 J=KI,ML
DO61I=1,120
LIG(I)=IRLANC
LF=IFREQ(J)/K
IF(LF.EQ.0)GOTO40
DO63I=1,LF
LIG(I)=IH
61 PRINT160,(J,IFREQ(J),(LIG(L),L=1,95))
62 CONTINUE
PRINT153
154 FORMAT(57X,'HISTOGRAMME DES FREQUENCES')
155 FORMAT(56X,28(1H-),///)
205 FORMAT(57X,'IMAGE GRADIENT AMELIORE CONTRASTEE')
206 FORMAT(56X,35(1H-),///)
151 FORMAT(57X,'UN H REPRESENTE ',I3,' VALEURS')
156 FORMAT(4X,'NIVEAU',2X,'FREQUENCE')
153 FORMAT(22X,98(1H-))
160 FORMAT(5X,I4,3X,I5,5X,1HI,1X,95A1,1HI)
RETURN
END

```

PROGRAMME FAISANT UNE CLASSIFICATION DE L'IMAGE

```
DEFINE FILE 5=CD(R:256,G:1,RW,BN,FI)
DIMENSION L(2)
DIMENSION LV(128),MA(128),MB(128),MC(128)
DATA IBLANC, IASTER/1H ,1H*/
DO 50 J=1,128
MA(J)=IBLANC
MB(J)=IBLANC
MC(J)=IBLANC
N=0
PRINT200
PRINT201
PRINT202
PRINT203
PRINT210
PRINT211
PRINT203
DO60 I=5272,5359
READ(5,1)(LV(J),J=1,128)
DO 70 J=1,128
IF(LV(J).EQ.10) GO TO 1
IF(LV(J).EQ.19) GO TO 1
IF(LV(J).EQ.25) GO TO 1
IF(LV(J).EQ.34) GO TO 1
IF(LV(J).EQ.39) GO TO 1
MA(J)=IBLANC
GOTO70
1
MA(J)=IASTER
70
CONTINUE
N=N+1
60
PRINT160,(N,(MA(J),J=1,128))
CONTINUE
PRINT204
160
FORMAT(1X,I3,128A1)
200
FORMAT(////////)
201
FORMAT(40X,'IMAGE DU GRADIENT AMELIORE CONTRASTE')
202
FORMAT(39X,46(1H-))
203
FORMAT(////////)
204
FORMAT(////////)
210
FORMAT(40X,'DETECTION DE CONTOURS')
211
FORMAT(39X,23(1H-),///)
STOP
END
```

