الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة الجامعات
Ministère aux Universites

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT D'ELECTRONIQUE.

PROJET DE FIN D'ETUDES

SUJET

ETUDE DE LA METHODE DES MOMENTS POUR LA RECONNAISSANCE DES FORMES APPLIQUEE AUX CARACTERES ARABES.

Proposé par : M^me = L. Hamami. Etudié par : M^eur = A. Merrouche Dirigé par : M^me = L. Hamami.

&

M^eur = S. Abdou.

PROMOTION

Juin 1991

E.N.P. 10, Avenue Hacen Badi - El-Harrach - Alger

ETUDE DE LA METHODE DES MOMENTS POUR LA RECONNAISSANCE DES FORMES APPLIQUEES AUX CARACTERES ARABES

Proposé par :
Mme HAMAMI

Etudié par :
Mr A. MERROUCHE
Mr S. ABDOU

DEDICACES

A la mémoire de notre regretté frère et ami
OUAHRANI MOHAMED

Je dedie ce travail :

A mes parents, pour les sacrifices consentis afin d'assurer mon avenir
A ma soeur
A mes frères
A ma famille
A mes amis

Soriane

Je dedie ce mémoire :

A mes parents
A mon frère MOHAMED
A mes frères et soeurs
A tous mes proches
A mes amis

Ahmed

REMERCIEMENTS

On ne saurait présenter la présente étude sans exprimer nos remerciements :

- A tous les professeurs de l'ECOLE NATIONALE POLYTECHNIQUE qui ont contribué à notre formation, en particulier Madame HAMMAMI qui a bien voulu diriger cette étude.

- Au personnel du centre de calcul de l'ENP, qui n'a jamais hésité de mettre à notre disposition les moyens nécessaires pour l'élaboration de ce modeste travail.

- A tous ceux qui ont participé, de près ou de loin, à la réalisation de ce travail et trouveront ici l'expression de notre profonde gratitude.

Nous remercions également tous les membres du jury, qui ont bien voulu accepter de juger ce travail

RESUMES

الموضوع: دراسة طريقة العزوم للتعرف على الاشكال مطبقة على الحروف العربية.
ملخص: في هذه الدراسة تم انجاز نظام للتعرف على الحروف العربية المعزولة. في المرحلة الأولى تم التطرق الى طرق معالجة الصور و ظبط محيطها. انطلاقا من احداثيات المحيط تم التحصل على سلسلة من العزوم. برهن ان هذه المجموعة من الخصائص كافية لتصنيف الحروف العربية المعزولة.

Title : *Recognition of isolated arabic caracters by moments invariants.*

Abstract : *In this study a recognition system has been developed for the recognition of isolated arabic caracters. In the first stage, methods of image processing and contour detection have been developped. After a successin of moments is obtained from the coordinates of contour points. This set of moments has proven to be sufficient for the classification of isolated arabic caracters.*

Titre : *Etude de la methode des moments pour la reconnaissance des formes appliquee aux caracteres arabes.*

Resume : *Dans cette etude un système de reconnaissance a été developpé pour la reconnaissance des caractères arbes isolés. Dans une première étape, des methodes de pretraitements et de detection de contours ont été developpées. En fin une serie de moments est obtenue à partir des coordonnées des contours. Il a été montré que cet ensemble d'attributs était suffisant pour la classification des caractères arabes isolés.*

SOMMAIRE

INTRODUCTION

La vision assistée par ordinateur (V.A.O), dans laquelle vient s'integrer le traitement d'image, s'est affirmée comme un domaine de recherche privilégié. Le sens de la vue est celui qui nous apporte la plus grande quantité d'information sur ce qui nous entoure, du fait de la multiplicité de ses dimensions: spaciale, énérgétique et temporelle. En effet, la vue constitue un moyen pour une machine (robot) de percevoir son environnement. Le problème de la vision consiste à identifier des formes visuelles simples à l'aide de methodes algorithmiques de reconnaissance de formes, par exemple caractères imprimés d'un texte, objets bien localisés d'une scène, zones particulières sur une image de satellite, etc... Néamoins, la comprehension veritable d'une scène ou d'une image nécessite d'aller au delà de ces methodes, en se fondant sur un ensemble de connaissances propre au domaine d'application envisagé. C'est le cas d'interprètation d'images radiologiques en vue d'un diagnostic médical, de la comprehension d'un texte imprimé ou manuscrit, du contrôle de la qualité d'un objet sur une chaine de fabrication, du guidage d'un véhicule autonome etc... .Le traitement d'image a été la première application de l'*intelligence artificielle* sur les sites indutriels.

Le terme V .A O symbolise toute la chaîne visuelle, partant de l'image brute jusqu'a l'interpretation de son contenu . Il y existe donc un aspect decisionnel très important lors de la phase de compréhension, d'où l'interaction avec l'*intelligence artificielle*. Le traitement d'image s'intègre dans ce processus comme un outil, il n'a aucun pouvoir décisionnel et se definit comme un ensemble de tâches déstinées à extraire de l'image des informations

qualitatives et quantitatives qui sont transmises au module de niveau superieur. Ceci permet d'utiliser des langages de programmation classiques tels que le 'Pascal' ou le 'C' pour réaliser des algorithmes d'imagerie. Le système d'interpretation étant en general developpé à l'aide de langages plus spécifiques tels que le Prolog ou Lisp.
La Fig 1 montre les composantes d'une chaine de reconnaissance visuelle.

Le capteur dépend du domaine d'application mais est essentiellement une caméra. Les prétraitements réalisent les tâches d'amelioration de l'image ainsi que la mise en evidence des *points interessants*. L'extraction des caracteristiques a pour rôle de faire une description de l'image compatible avec la description des objets contenus dans la base de connaissances, celle-ci constitue la *'memoire'* du systeme de vision. Le terme d'objet est utilise ici au sens large, dans notre cas il designe un caractère. Enfin le systeme decisionnel, par comparaison des caracteristiques qui lui sont fournies et le contenu de la base de connaissance va effectuer la reconnaissance proprement dite. Il apparait clairement que l'extraction de caracteristiques joue un rôle d'interface et est directement liée au choix du type de fonctionnement du système. Ce dernier caracterise l'aspect intelligent du systeme de vision par ordinateur.

DOMAINES D'APPLICATION DE LA R.F :

1 Reconnaissance des signaux d'origine humaine :

L'exemple le plus célèbre est celui de la parole dont les performances sont actuellement au stade de réalisation de

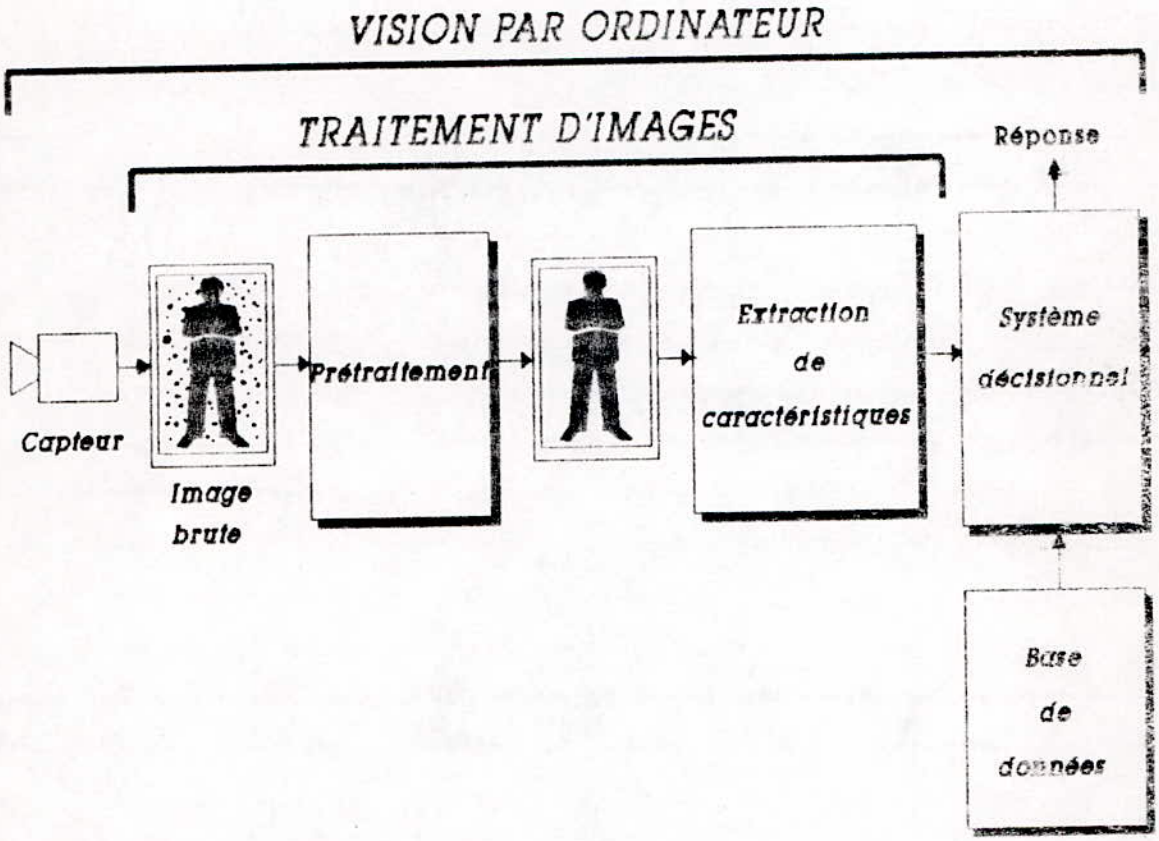


Fig 1. *La chaine de V . A . O*

machine d'usage public capable de reconnaitre des suites de mots choisis parmi plusieurs. Il ya aussi la reconnaissance du locuteur qui connait de larges applications dans le domaine de la sécurite.

La RF a apporté une aide considerable dans les divers domaines notamment en médecine pour les diagnostics et depouillements des signaux medicaux, tels l'éléctrocardiogramme l'éléctro-encephalogramme, l'éléctrorétinogramme, l'analyse des chromosomes, mesure des debits sanguins, application à la psychologie. Cependant les applicatins les plus avancées concernent les images biomedicales : l'analyse des cellules sanguines, la cytologie (étude des ségments d' A.D.N, comptage globulaire, étude des structure des virus, enzymes, comptage des grains de pollen), des cellule cérébrales, ...etc .

1.1 La reconnaissance des caractères (lecture optiques) :

Bien que les traitements d'information sont de plus en plus rapides, la saisie des données constitue un véritable obstacle aux performances des machines.

C'est pour éviter la création d'ateliers de saisie démesurés, que dés les années soixante, la reconnaissance optique des marques puis des caractères est apparu comme solution naturelle à l'automatisation des entrées massives de données.Cet autre signal important qu'est l'écriture est l'un des champs favori de la R.F, bien que les résultats restent médiocres comparés à la quantité de recherches effectuées dans ce domaine.

La complexité du problème réside dans l'infinie variabilité de l'écriture manuscrite. Cette variabilité constitue la cause essentielle des semi-échecs récoltés. Même l'être *intélligent* considéré comme modèle idèal n'est pas infaillible : il commet 4% d'erreurs à la lecture d'un manuscrit à l'abscence de contexte.

2 Reconnaissance des signaux d'orgine naturelle :

Les mesures faites sur la terre, les photos satellites, les contrôles sismologiques, mesures de houle marine sont des exemples de tels signaux. Les applications sont diverses :

- Détection des ressources, météologie, prévision des phénomènes naturels, classement sur les images satellites (recherche des formes de routes, analyse de côtes marines et des rivières, etc...).
- En astronomie : connaissance des étoiles, etude des atmosphères planètaires, etc... .
- En physique des hautes énergies, les images de chambre à bulles sont traitées pour détecter et localiser des collisions particulières entre particules élémentaires, reconstitution des particules de trajectoires, etc...
- En industrie :
  * Vision robotique (reconnaissance des pièces mécaniques, analyse des composants organiques dans les produits alimentaires, analyse des circuits integrés(V.L.S.I), contrôle de soudures, diagnostics de pannes, surveillance).
  * Télédetection : identification des espèces végétales, reconnaissances des rôches etc...

CHAPITRE I

CONFIGURATION MATERIELLE

Bien qu'étant encore au stade de deve loppement, surtout au niveau industriel, le traitement d'image en général et la reconnaissance de formes en particulier commence à voir apparaitre nombre de machines spécialisées. Celles-ci offrent l'avantage d'une architecture optimisée pour l'imagerie, ce qui leur procure une grande rapidité de traitement. Ces machines feront l'objet de ce chapitre.

1. DESCRIPTION D'UN MATERIEL PROFESSIONNEL :

Sa complexité peut-être très variable selon le domaine d'application. On peut pour cela séparer les utilisations du traitement d'images en deux domaines : celui où la contrainte du temps réel doit être impèrativement satisfaite, et celui où le temps n'est pas un élément fondamental. La Fig I.i illustre les differents éléments d'un système professionel minimal, déstiné au développement d'applications basées sur le traitement d'image. Examinons un par un chacun de ces constituants :

1.1. La caméra :

- Les cameras à tubes :

Elles sont composées d'une cible photoconductrice qu'explore un faiseau éléctronique. L'exploration s'effectue ligne par ligne. A chaque ligne correspond en sortie un signal analogique proportionnel à l'intensité.

- Les caméras CCD (Charge Coupled Device) :

Celles-ci sont constituées par un assemblage de photodiodes chacune d'elles delivrant une intensité

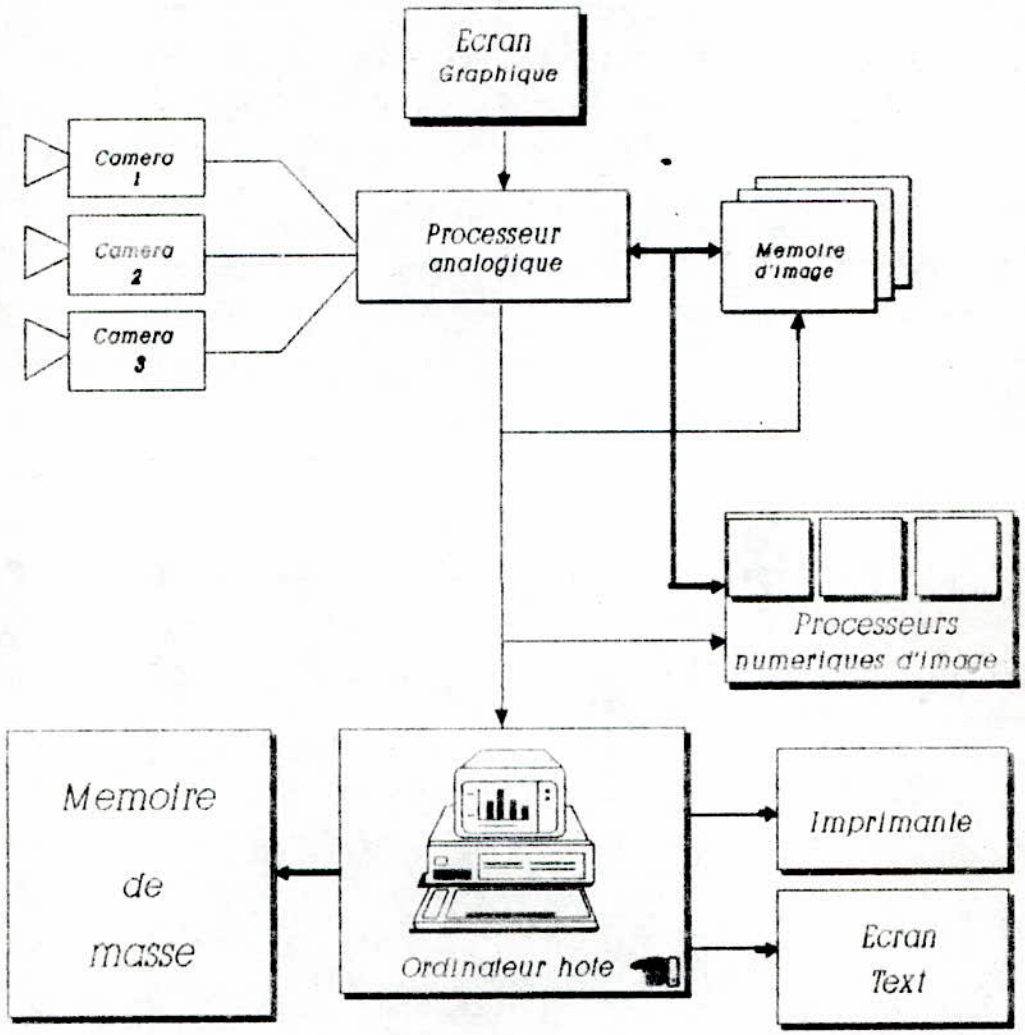


Fig I.1. *Configuration matérielle d'un système professionnel.*

proportionelle à un point de l'image appelé *pixel*. Contrairement aux caméras à tubes qui fournissent un signal continu qu'il faut échantillonner spacialement pour isoler chaque pixel, les cameras CCD fournissent directement l'intensité pour chaque point de l'image puisque à chacun correspond une photodiode. On distingue deux familles de cameras CCD :

* *Les cameras matricielles* : Les photodiodes sont assemblées en une matrice de n lignes et p colonnes. Cette matrice peut-être carrée, les résolutions courantes sont alors *100x100* ou *256x256*, ou bien rectangulaires(*300x512*), ce qui correspond au rapport de *4/3* d'un écran de télèvision.

* *Les cameras unilignes* : Comme leur nom l'indique, les photodiodes y sont agencées suivant une seule ligne. Elles ont l'avantage de posséder une resolution superieure(*2048 points*). L'obtention d'une image necessite cependant le déplacement de la camera afin d'acquerir plusieurs lignes, pour obtenir une image bidimensionnelle. Ce type de camera est bien adapté à l'acquisition des documents; on dèplace la feuille à vitesse constante perpendiculairement à la camera et on fait une acquisition d'image suivant une periode T.

1.2 Le processeur analogique :

Cet élément peut être considéré comme gestionnaire des entrées-sorties du système. Ses fonctions sont multiples :

- Sélection de la camera active lorsque le système comporte plusieurs entrées et étalonnage du signal.
- Selection de l'écran d'affichage si le système comporte plusieurs sorties.

- Acquisition du signal video effectuée par le digitaliseur: cet élément est fondamental, c'est en effet lui qui réalise l'échantillonage spacial et le codage de l'image. L'échantillonage correspond au découpage du signal en pixel. Nous pouvons constater que celui ci est dependant du type de cameras utilisées. Le codage correspond à la quantification de l'intensité de chaque pixel en une valeur numérique couramment appelée *niveau de gris (Gray level)*. Des codages classiques se font sur *16*, *32*, ou *64* bits, mais pour des raisons technologiques la valeur la plus utilisée est de 256, bien que l'œil humain ne soit pas capable de percevoir autant de nuances.

- Prétraitement de l'image lors de l'acquisition. Il s'agit généralement des opérations ponctuelles qui visent à améliorer le contraste de l'image.

1.3. La mémoire d'écran :

L'image, une fois numérisée, se présente sous la forme d'un tableau I de n lignes et p colonnes. Chaque élément $I(n,p)$ représente un pixel de l'image et à sa valeur est associée l'intensité du point qui est en général *0* pour le noir et N pour le blanc, avec N = *255* le plus souvent. Ce choix de *256* niveau s'explique par le fait que ce type de codage nécessite *8* bits par pixel, ce qui est une valeur commode. La mémoire d'écran se présente donc sous la forme d'une matrice de $n \times p$ cases de K bits chacune avec le plus courament $n=p=256$ ou *512*. Cette dernière valeur est la plus rencontrée car elle représente un compromis acceptable entre la définition spaciale de l'image et les temps nécessaires à son traitement. En fait, la mémoire d'image se résume rarement

à un seul tableau. En effet, il est pratique, parfois même indispensable de disposer de plusieurs matrices $I_1, I_2, \ldots, I_N$ pour effectuer un traitement. Par exemple dans I_1 se trouve l'image originale, dans I_2 une image intermédiaire et dans I_3 l'image résultat.

Les échanges de données entre la mémoire d'image, le processeur analogique et le ou les processeurs spécialisés s'effectuent par l'intermediaire d'un bus rapide.

1.4. Le processeur d'image :

Sous cette appelation se groupent un ou plusieurs processeurs spécialisés chacun pour une opération bien précise. On trouve des processeurs de filtrage, d'extraction de contours, d'opérations morphologiques etc... . Une grande partie des algorithmes decrits dans cette étude sont disponibles sous forme *hardware*. Ils ont été conçus de manière à réaliser leurs traitements à grande vitesse : de l'ordre de la fréquence image, soit *1/25* de seconde.

1.5. L'ordinateur hôte :

Il contrôle le fonctionement décrit précedamment.

- *Le processeur analogique* : Choix de la camera, choix de l'écran de visualisation, commande d'acquisition d'image vers la mémoire d'image selectionnée.
- *Les processeurs spécialisés* : Choix, paramétrage et lancement de l'algorithme à appliquer.
- *Les échanges mémoire de masse - mémoire d'image* : Sauvegarde d'images sur disque et vice versa.

Dans le cadre d'une application industrielle,

l'ordinateur hôte a pour rôle de piloter le systéme de traitement et de recueillir les informations issues de celui-ci afin d'engendrer la commande ou la decision appropriée.

Dans le cadre du developpement d'un algorithme originale l'ordinateur hôte s'occupe de traitements. Dans ce cas la présence de processeurs numeriques d'images dans le systéme n'est pas nécessaire.

1.6. La mémoire de masse :

Une image noir et blanc *512x512* sur *256* niveaux occupe *256 Ko*, la même image en couleurs en occupe trois fois plus. Il est donc nécessaire de disposer d'une mémoire de masse importante.

2. CONFIGURATION DU SYSTEME UTILISE :

Il n'est cependant pas impératif de posséder le matériel décrit précédement pour s'initier au traitement d'image pour la reconaissance de forme. Cela en pose évidemment quelques limitations dans la représentation et la taille des images à traiter mais n'enlève rien quant à l'etendu des problémes posés à leur compréhention.

2.1 Description matérielle :

Compte tenu des moyens disponibles, le matériel utilisé se compose de :

- Un micro-ordinateur de type compatible IBM PC/AT de *640 Ko* de mémoire RAM.

- Une carte graphique VGAHi permetant une résolution de 640x480 sur 16 niveau de gris. Le rôle de la mémoire d'ecran est joué par la carte graphique.
- Un ecran couleur ou monochrome.
- Une unité de disquettes (3"1/2 ou 5"1/4), un disque dur.
- Une imprimante graphique permetant de garder la trace des images traitées afin de comparer les résultats.

2.2. Organisation logitielle :

Les programmes décrits ont été developpés en Turbo Pascal 5.5. Ce langage s'est avéré puissant en temps de calcul et offre un graphisme de qualité.

Nous avons opté pour des images de taille 128x128, et ce pour deux raisons :

- Leur taille est modérée (16 Ko), ce qui permet de garder des temps de calcul raisonnables et rend possible l'utilisation de disques souples comme mémoire de masse.
- Il est possible d'afficher plusieurs images simultanément :
  * Carte CGA :Deux images.
  * Carte EGA :Huits images.
  * Carte VGA :Douze images.

On peut ainsi comparer le résultat d'un algorithme par rapport à l'image originale.

Nous pouvons remarquer l'absence dans notre systéme d'un digitaliseur. Il est donc impossible de travailler sur des images réelles. Ceci est loin d'être un inconvenient pour un apprentissage. De même, tout algorithme d'imagerie se doit d'être experimenté au début sur des images simples et dont on connait les caractéristiques. Cela permet d'analyser sans

ambiguïté les effets de l'opérateur. On utilise pour cela des images lues par scaner ou synthétiques générées à l'aide d'un éditeur graphique dont le principe est illustré par la procedure *CreeImage*.

Des procedures ecrites en turbo pascal ont été développées illustrant tous les algorithmes qu'on étudiera. Ces dernières, ainsi que d'autres utilitaires, ont été intégrées dans un menu graphique. Cet environnement logiciel utilise le concept de fichier de travail.

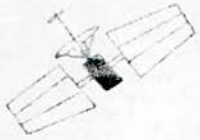
Les images générées par la procedure *CreeImage* sont sauvegardées dans des fichiers dont l'extension est ".*LET*". A la fin d'une transformation, l'image résultat garde le même nom avec une nouvelle extension représentant les trois premières lettres du nom de la transformation. Un manuel d'utilisation est donné en annexe 2. De manière analogue, pour le calcul des moments, le résultat est sauvegardé sous le même nom de fichier mais avec une extension ".*MOM*".

Les constantes utilisées par le logiciel sont les suivantes :

* *NIVMAX* : Correspond au niveau de gris le plus élevé que l'on peut avoir, soit *15* pour coder une image sur *16* niveaux.

* *MAXLIN* : Numéro de ligne le plus élevé dans l'image, soit *127* pour une image *128x128*.

* *MAXCOL* : Numéro de colonne le plus élevé dans l'image, soit *127* pour une image *128x128*.



CHAPITRE II

PRETRAITEMENTS
AMELIORATION D'IMAGE

Aprés digitalisation l'image brute acquiert une structure matricielle simple dans son organisation, mais complexe dans son contenu. Ceci est dû, d'une part à la grande quantité d'information qui y reside, d'autre part au grand nombre de processus, indépendant de l'utilisateur, qui tendent à y introduire des distortions indésirables donc à la dégrader. Parmi les éléments de perturbation on a :

- L'eclairage qui peut-être trop puissant et sature la caméra, ou bien trop faible et rendre l'image trop sombre.
- La qualité de la prise de vue.
- Le bruit electronique engendré par le systéme de prise de vue.
- Les défauts du systéme d'échantillonnage et de quantification.

L'amélioration d'image (*Image enhancement*) consiste en un ensemble de méthodes destinées à ameliorer l'aspect visuel d'une image. L'ensemble des techniques ayant cette fonction fait de ce que l'on appelle :*Les prétraitements*, qui regroupent un ensemble de méthodes qui visent à attenuer les effets indesirables. Deux principales approches ont été envisagées pour améliorer une image.

- Les opérations basées sur l'examen de l'histogramme. Elles améliorent les défauts de prise de vue en jouant sur la dynamique de l'image.

- Le filtrage dont le but est de minimiser l'influence du bruit dans l'image.

1. MODIFICATION DE LA FORME D'HISTOGRAMME :

Dans une image naturelle qui a été quantifiée de maniére lineaire, une majorité de pixels ont une valeur inférieure à la luminance moyenne. C'est pourquoi les détails dans les regions sombres sont difficilement perceptibles. Il est souvent donc necessaire d'ameliorer le contraste (accentuation des contours) et la dynamique de l'image; l'histogramme se préte à des opérations ponctuelles sur les pixels qui permettent des corrections artificielles, efficaces et spectaculaires.

1.1. Egalisation d'histogramme :

Elle consiste en une transformation qui recadre l'image de maniére que l'histogramme de l'image résultat soit le plus proche possible d'une distribution uniforme.

Considerons l'image digitalisée sur N_I niveaux de gris, comportant p^2 pixels. Chaque pixel peut prendre une valeur aléatoire X_k de l'ensemble des niveaux de gris.

La fonction de répartition de X_k, notée $F(X_k)$ est definie comme etant la probabilité P_k pour que X_k prenne une valeur inférieur à a :

$$P_k(X_k<a) = F(a)$$

L'histogramme discret $P_k(X_k)$ (Voir Démo.II.1.a) est donné par :

$$Pk = \frac{F(Xk) - F(Xk-1)}{p^2} = \frac{Nk}{p^2}$$

où: Nk est le nombre de pixels au niveau Xk,
p^2 est le nombre total de pixels dans l'image.

L'histogramme cumulé discret (Voir **Démo.II.1.b**) est donné par :

$$Hk = \frac{\sum_{i=0}^{k} ni}{p^2} \qquad [ref\ 2]$$

Si l'image résultat est codée sur NF niveaux de gris, alors le nombre de pixels pour chaque niveau de l'histogramme égalisé ideal est constant, et est égale à p^2/NF (car on a p^2 pixels dans l'image). Il est evident que le nombre de niveaux final NF doit-être inférieur au nombre de niveaux initial NI. (En général en prend $NF=NI/2$, qui est l'intégration minimale). L'histogramme final est donc séparé en bandes de taille NI/NF.

1.1.1. Méthode d'egalisation :

- Partant d'un niveau zéro, on calcule la valeur cumulée des niveaux suivants dans l'histogramme original, jusqu'à ce que la somme soit la plus proche possible de la valeur moyenne ideale p^2/NF.
- Les niveaux de l'histogramme initial qui ont participé à cette somme sont remplacés par un niveau unique qui se trouve au centre de la premiére bande de l'histogramme final.
- On recommence en suite l'opération pour les bandes

suivantes. Notons que plus le nombres des niveaux N_F finaux est restreint, plus l'intégration est importante du fait de la valeur plus élevée de P^2/N_F.

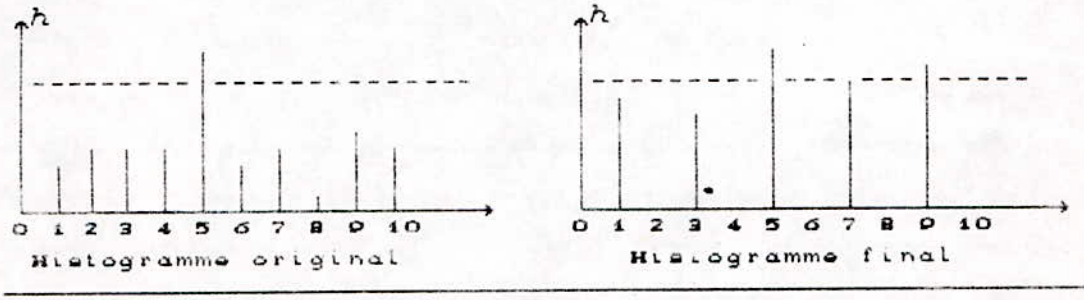


Fig II.1.*Principe de l'egalisation d'histogramme.*

1.1.2. Organigramme de la méthode :

Variables:

Ima :Image à egaliser.
Histo :Histogramme de l'image.
Transfo :Table d'egalisation.
iDebut :Fixe le début de la zone d'intégration sur Histo.
Ifin :Fixe la fin de la zone d'intégration.
Bande :Largueur de la bande finale N_I/N_F.
CentreBande :Fixe le centre de la bande courante.
Cumul :Somme des niveaux de gris.

Le shéma de l'organigramme est sur la Fig II.2.

1.2. Autre modification d'histogramme :

L'egalisation n'est pas la seule technique de transformation d'histogramme, on peut imaginer des transformations d'histogramme qui recadrent l'image de maniére que l'histogramme final se raproche d'une forme exponentielle

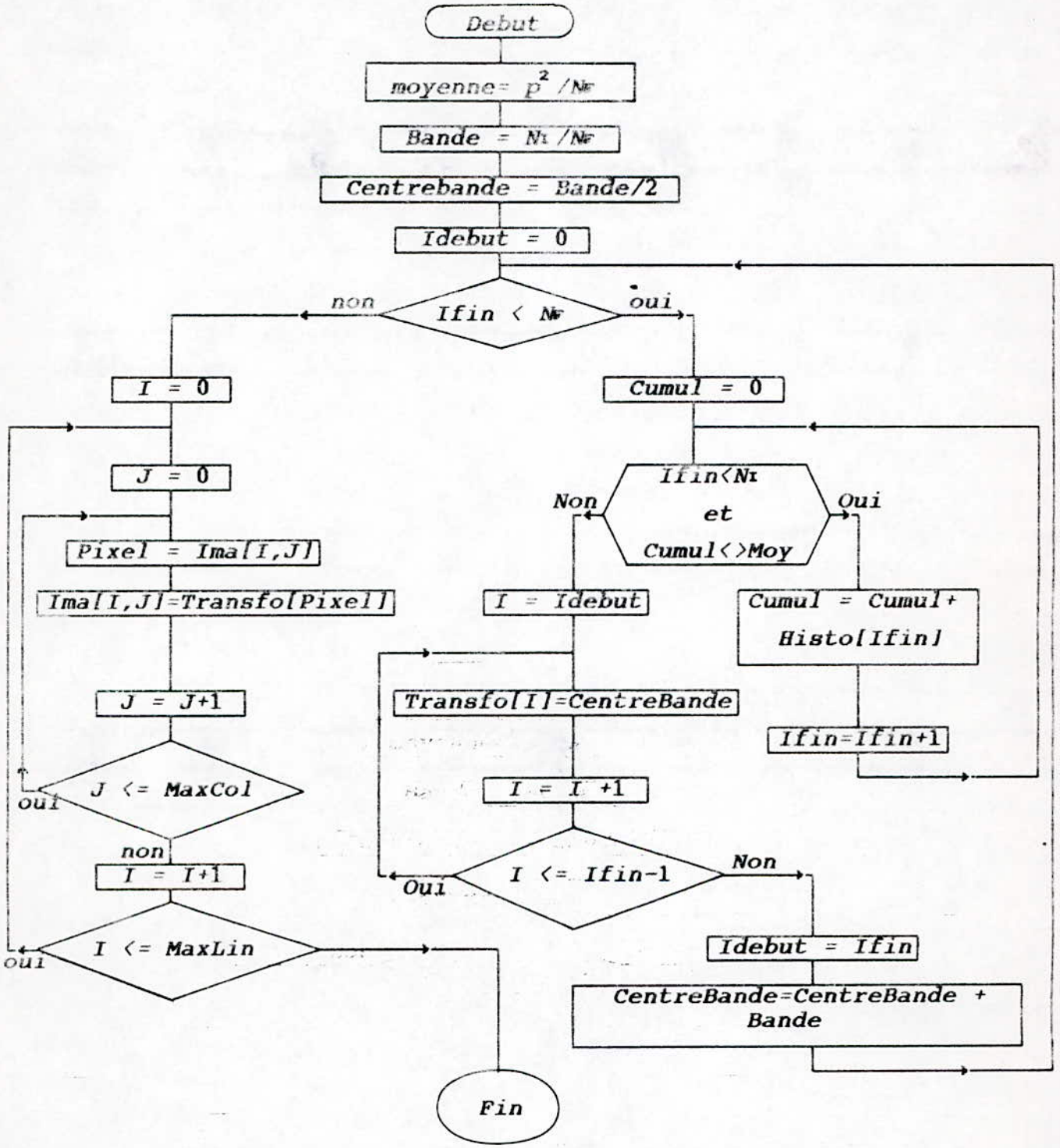


Fig II.2. *Organigramme de la methode d'égalisation.*

ou hyperbolique. Dans ce cas, la méthode décrite reste valable, les seuls changements interviennent dans le calcul de la valeur ideale. Celle ci, au lieu d'être constante et egale à P^2/N_F, devient variable et suit l'équation de la courbe désirée.

1.3. Conclusion sur la méthode des histogrammes :

L'égalisation est la méthode la plus employée pour améliorer le contraste et renforcer la dynamique de l'image, les contours sont accentués, mais en même temps l'image prend un aspect dur et les bruits sont renforcés.

L'aspect dur de l'image égalisée est dû à la présence de fortes différences entre les diférents pics de l'histogramme original. En effet, lorsqu'un pic est largement supérieur à la valeur moyenne idèale, il ne peut être divisé pour occuper plusieurs régions de maniére arbitraire. Cela n'aurait pas de sens car cela signifierait qu'un niveau de gris initial pourrait avoir plusieurs niveaux de gris finaux. Ce pic de grande taille concerne donc un nombre de pixels qui pourraient occuper plusieurs bandes et qui finalement n'en occupe qu'une, d'où une contraction vers la gauche (zones sombres) de l'ensemble des pics suivants.*[ref 2]*

Dans le cas d'image binaire, l'égalisation n'apporte pas de changements.

2. LE FILTRAGE :

2.1. Introduction :

2.1.1. Le bruit, ennemi du signal image :

Dans de nombreux domaines scientifiques, le bruit joue un rôle fondamental. Il est à l'origine d'un grand nombre de difficultées lorsque l'on desire analyser un quelconque signal. Le bruit dans une image est le résultat du bruit electronique du capteur et de la qualité du digitaliseur. L'existence du bruit n'est pas seulement liée au matériel utilisé, mais aussi à l'image elle même : zones de fort contraste, parties plus ou mois sombres , etc...

Dans une premiére approximation, on peut caractériser le bruit par un bruit dit *impulsionnel*. C'est à dire un bruit affectant de brusques variations des pixels isolés. Ce modéle permet de gérer de maniére simple du bruit dans une image et a l'avantage d'être aisé à programmer en utilisant une fonction aléatoire paramétrée en amplitude et en densité. La densite va déterminer la propotion moyenne de pixels atteints par le bruit, l'amplitude va en déterminer le niveau. C'est le principe de la procedure *IntroBruit*. Celle ci permet de donner un aspet plus realiste à des images synthétiques et est trés utile pour évaluer les performances des algorithmes de filtrage que nous étudierons.

2.1.2. Le filtage :

Nous avons vu dans le paragraphe précédent, des techniques qui visaient à améliorer le contraste de l'image. Elles portaient le nom de *Transformations Ponctuelles* car chaque pixel était transformé indépendamment de la valeur de ses voisins.

Or l'image possède une certaine redondance spaciale : des pixels voisins ont généralement ou pratiquement les mêmes caractéristiques.

Nous avons defini le bruit comme un phénomène de brusques variations d'un pixel isolé par rapport à ses voisins.

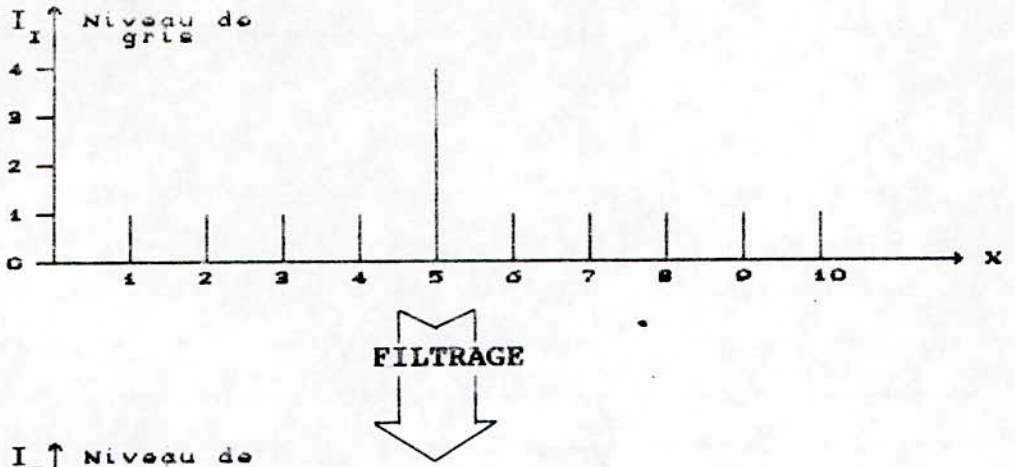
Ces considérations nous amènent à déduire que pour lutter contre les effets du bruit, il est nécessaire d'opèrer des transformations qui pour chaque pixel tiennent compte de son voisinage. Dans ce but plusieurs techniques dites *Filtrage de l'image* sont utilisées. Le bruit considéré apporte une importante contribution au spectre hautes fréquences d'où l'utilisation de filtre passe bas. On distingue deux types de filtrage passe bas :*Le filtrage lineaire* où la transformation d'un pixel est le fruit d'une combinaison lineaire des pixels voisins, et *Le filtrage non lineaire* où les pixels voisins interviennent suivant une loi non lineaire.

2.2. Le filtrage lineaire :

2.2.1. Introduction :

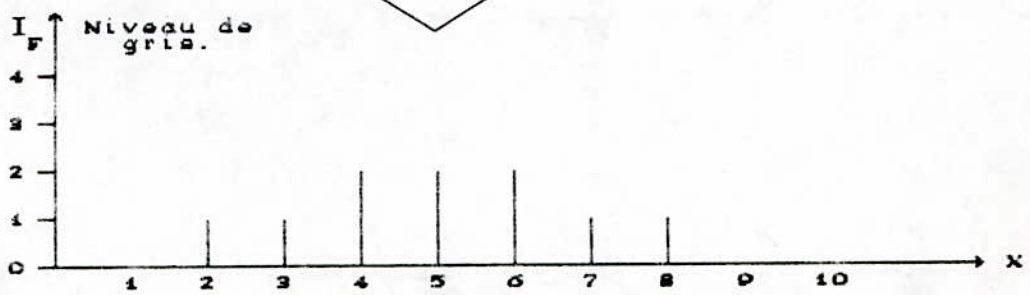
On désire filtrer une image bruitée, c'est à dire, si ce n'est éliminer complètement le bruit, du moins en diminuer sensiblement les effets. Une méthode simple consiste à considerer chaque point de l'image et d'en faire la moyenne avec les huit pixels qui lui son voisins comme indiqué sur la Fig II.3 :C'est le principe du *filtre moyen*.

D'une façon générale, le filtrage spacial consiste en une convolution de l'image initiale I_I avec un masque carré

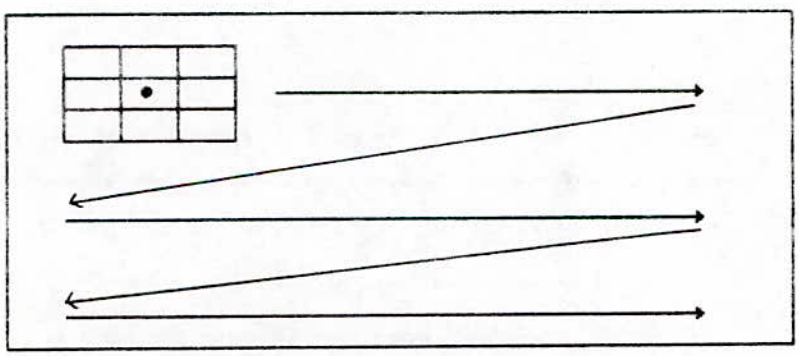


FILTRAGE

.a.



$$I_F(x) = \frac{I_I(X-1) + I_I(X) + I_I(X+1)}{3}$$



.b.

$$I_F(X,Y) = \frac{1}{9}\left[I_I(X-1,Y-1)+I_I(X-1,Y)+I_I(X-1,Y+1)+I_I(X,Y-1)+ I_I(X,Y)+I_I(X,Y+1)+I_I(X+1,Y-1)+I_I(X+1,Y)+ I_I(X+1,Y+1)\right]$$

Fig II.3 : *Principe du filtrage par moyenne.*
(a) Cas unidimensionnel, (b) Cas bidimensionnel

dont les coefficients déterminent la nature du filre utilisé. Partant d'une image initiale I_I sur laquelle on veut opérer un filtrage à l'aide d'un masque H de taile $(n+1)(n+1)$, dont la somme des coefficients est K, l'image finale s'écrit pour tout point x,y :

$$I_F(x,y) = \frac{1}{K} \sum_{i=-n/2}^{n/2} \sum_{j=-n/2}^{n/2} H(i+n/2, j+n/2).I_I(x+i, y+j)$$

2.2.2. Notion sur la convolution :

La formule précédente n'est autre que l'expression, en chaque point de l'image, d'un *produit de convolution discréte* et l'on peut écrire pour l'image entière.

$$I_F = H \otimes I_I$$

où "⊗" symbolise l'opérateur de convolution.

H, est connu en traitement de signal comme etant ce que l'on appelle '*La réponse impultionnelle du filtre*'. Celle ci correspond à l'image obtenue en appliquant le filtre à une image composée d'un seul pixel sur fond noir (*Impulsion de DIRAC* :élément neutre de la convolution).

Nous venons de voir que le filtage consiste en la convolution de l'image avec un masque appelé *noyeau (Kernel)*. La convolution est une notion trés importante en traitement d'image, en particulier du fait d'une de ses propriétés concernnat la transformée de Fourrier.

Rappelons la définition de celle ci dans le cas continu

bidimentionnel:

$$g(Wx, Wy) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} G(x, y) \exp(-j(W_x x + W_y y)) \, dx \, dy$$

$$= F\left[G(x, y)\right]$$

où Wx, Wy sont les fréquences spaciales et 'j' l'imaginaire pur de carré -1.

Une propriéte fondamentale du produit de convolution par rapport à sa transformée de Fourrier est :

$$F\left[H(x, y) \otimes I(x, y)\right] = F\left[H(x, y)\right] \; F\left[I(x, y)\right]$$

Donc la transformée de Fourrier d'un produit de convolution est egale au produit simple des transformées de Fourrier. On voit donc qu'un filtrage peut aussi être réaliser de la maniére suivante, si H est le noyau et I l'image :

- On effectue séparement les transformées de Fourrier de H et de I .
- On multiplie ces deux tansformées.
- On applique au produit la transformée de Fourrier inverse, on obtient ainsi l'image filtrée.

Ce type de filtrage dit frequentiel, revient chér car il faut, par transformée de Fourrier, passer de I, H à i, h, et par la transformée de Fourrier inverse passer du produit (i.h) à l'image finale I_F. *[ref 2]*

2.2.3. Effet de bord :

Jusqu'à présent nous n'avons pas parlé des problèmes liés au bords de l'image. En effet, si l'on applique par exemple un filtre H au point de l'image de coordonnées $(0,0)$, nous voyons que ceci n'est pas possible puisque le filtre fait alors appel à des points de coordonnées negatives qui n'appartiennent pas à l'image. De manière generale, pour un filtre de taille $(n+1)(n+1)$, il y atout autour de l'image un cadre d'épaisseur $n/2$ où le filtre n'est à priori pas applicable (Voir Fig II 4.a). Les différentes possibilitées sont alors :

- Filtrer uniquement la partie interieure de l'image variant de $n/2$ à $p-n/2$. L'image résultat est alors de taille $(p+1-n)(p+1-n)$. C'est la possibilité que nous avons utilisée pour nos procedures de filtrage.
- Réaliser un effet dit de miroir. On ajoute autour de l'image initiale un cadre d'épaisseur $n/2$ dans lequel on recopie les $n/2$ lignes du bord de l'image comme illustré sur la Fig II.4.b. Dans ce cas l'image résultat est de taille $p \times p$.
- On reprend la méthode précèdente mais on remplit le cadre de 0. Voir Fig II.4.c.

2.2.4. Filtre moyen :

2.2.4.1. Principe :

Ce type de filtre utilisant la moyenne non pondérée des voisins peut être mis sous la forme d'un masque tel que celui

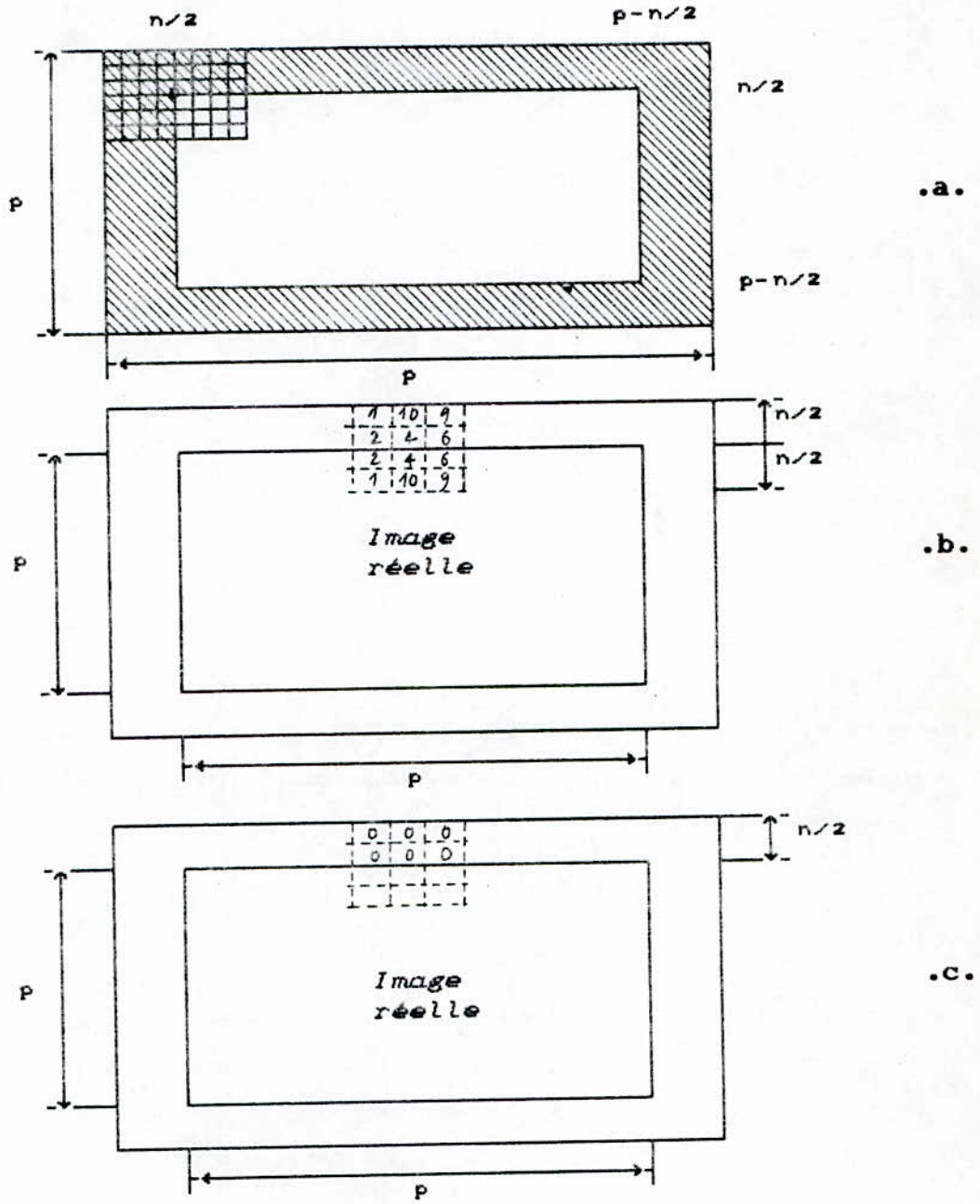


Fig II.4. *Résolution des effets de bords*
(a) Domaine d'utilisation du filtre, (b) Effet de miroir,
(c) Mise à zéro de la courone.

ci :

$$H_1 = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

On va alors deplacer ce masque sur toute l'image, le pixel affecté par la transformation étant le pixel central du masque. Le facteur 1/9 est égal à la somme des coefficients du masque et sert à normaliser le filtre de manière que celui ci n'influe pas sur l'intensité globale de l'image. Nous avons alors pour chaque pixel de coordonnées x,y :

$$I_F(x,y) = \frac{1}{9}\sum_{i=-1}^{1}\sum_{j=-1}^{1} H_1(i+1,j+1)\ I_i(x+i,y+j)$$

Ce filtrage est la technique la plus directe pour adoucir une image trop heurtée dans ses contours par manque de dégradés ou abimée par un bruit à haute fréquence. Mais crée du flou sur les bords d'objets et autres détails à fort gradient d'intensité.(Voir Démo.II.2)

D'autres filtres on été réalisés en utilisant des coefficients de pondération différents. Par exemple :

$$H_2 = \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad H_3 = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Le filtre H_2 donne plus de poids au pixel central, ce qui le rend moins actif que le filtre H_1, le filtre H_3 privilégie

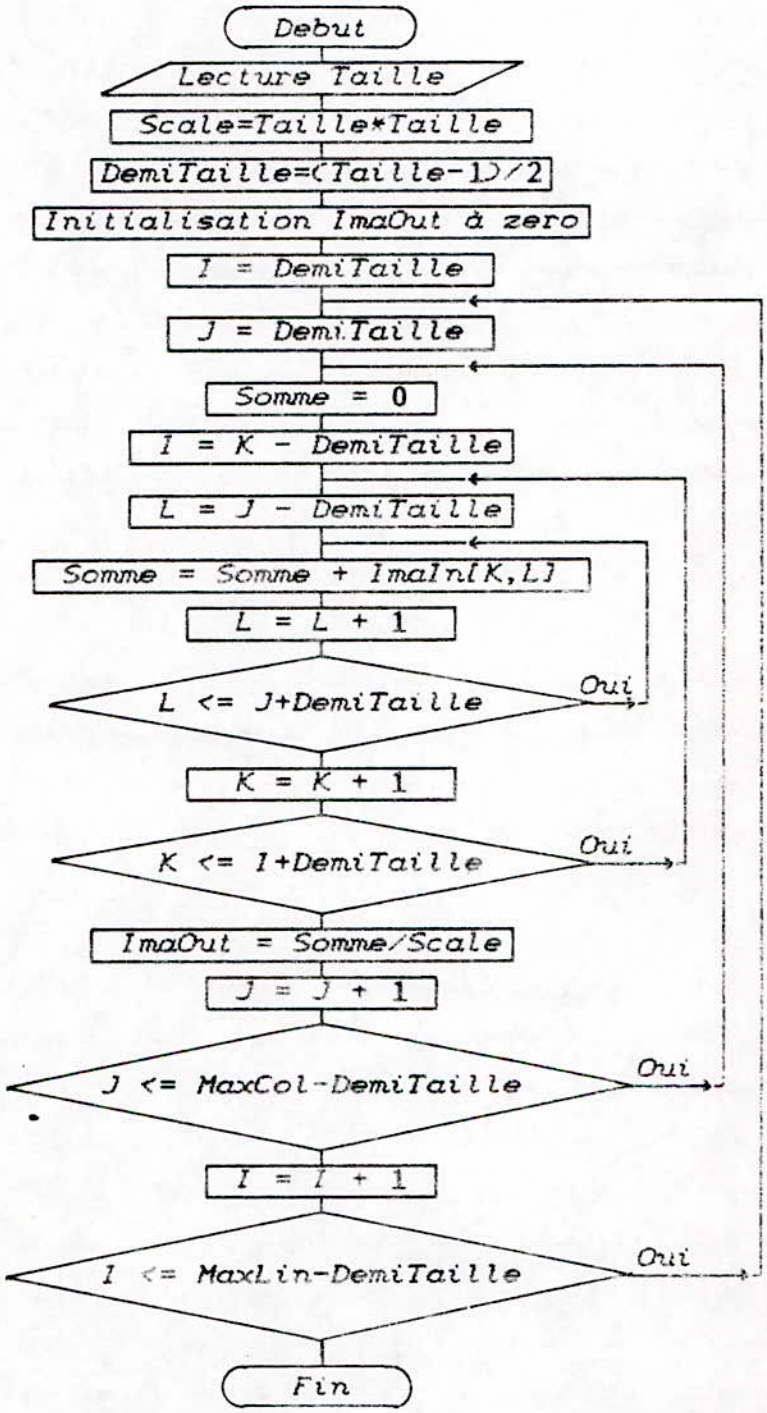


Fig II.5. *Organigramme du filtre moyen.*

les directions X et Y. Plus la taille du filtre est large plus le filtrage est important mais aussi plus l'image filtrée perd de sa netteté.*[ref 8]*

2.2.4.2. Organigramme :

Variables :

ImaIn	:Image Initiale.
ImaOut	:Image finale.
Taille	:Taille du filtre.
DemiTaille	:Demi taille du filtre.
Scale	:Somme des coefficients du filtre.
Somme	:Somme des produits.

Le shéma de l'organigramme est sur la Fig II.5.

2.2.5. Filtre de Gauss :

2.2.5.1. Pricipe :

Il existe une grande variété de filtres plus au moins utilisés, et il serait fastudieux de vouloir tous les décrire.Le filtre de *Gauss* représente l'un des filtres linéaires les plus utilisés, vue sa facilité de mise en œuvre et la bonne qualité de ses résultats. Ce filtre tire son nom de la valeur de ses coefficients qui sont ceux d'une courbe de Gauss à deux dimensions.

Rappel : *courbe de Gauss à une dimension* :

L'equation de la courbe de Gauss s'écrit *[ref 6]* :

$$G(x,\sigma) = 1/\left(\sqrt{2\pi}\ \sigma\right)\exp\left(-\frac{x^2}{2\sigma^2}\right)$$

Cette courbe est positive, symetrique (Voir Fig II.6), et posséde les proprietés suivantes :

$$\int_{-\infty}^{+\infty} G(x,\sigma)\,dx = 1$$

$$\int_{-2\sigma}^{+2\sigma} G(x,\sigma)\,dx = 0.95$$

$$\int_{-3\sigma}^{+3\sigma} G(x,\sigma)\,dx = 0.999$$

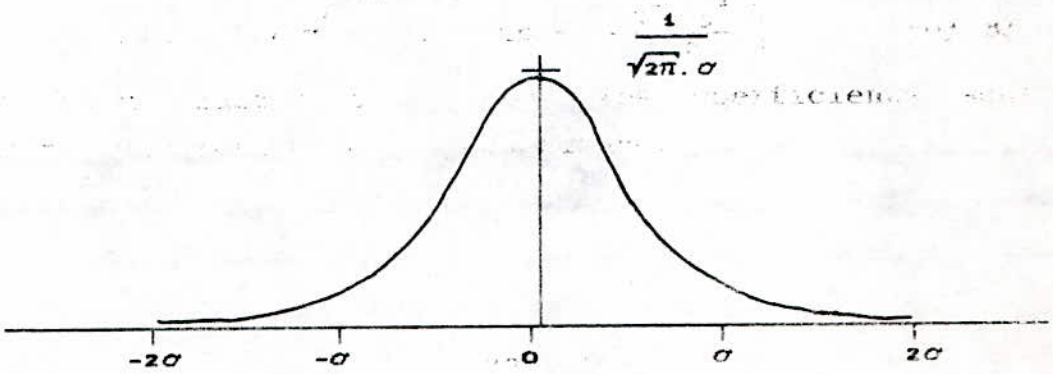


Fig II.6. *Courbe de* **Gauss** *à une dimention.*

Cas bidimentionnel - Calcul du filtre

Un filtrage gaussien consiste en la convolution d'une image I_I avec une gaussienne $G(x,y,\sigma)$:

$$I_F = I_I \otimes G$$

avec

$$G(x,y,\sigma) = 1/\left(\sqrt{2\pi}\ \sigma\right) \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \qquad \textit{[ref 8]}$$

où G est un masque carré dont les coefficients sont les éléments discrétisés de la gaussienne.

Comme le masque est de taille $(n+1)(n+1)$ finie et que la courbe de Gauss est definie de moins l'infini à plus l'infini, il est necessaire d'en prendre une portion finie. Or nous avons vu dans le précedent paragraphe que pour une largeur de 4σ, nous avions 95% de la courbe, et 99.9% pour 6σ. Il faut donc, pour obtenir une bonne approximation choisir un masque G de taille N egale à au moins 4σ.

Un rapide calcul montre que pour un masque de taille NxN pixels, il faut realiser $N^2 = 36\sigma^2$ multiplications, et $N^2-1 = 36\sigma^2-1$ additions par point de l'image pour effectuer la convolution. Donc le temps de calcul est proportionnel au carré de la taille du filtre, ce qui peut devenir prohibitif pour un fitre de grande taille.

Or la fonction $G(x,y,\sigma)$ est séparable, c'est à dire qu'elle est réalisable soit par une convolution par un filtre NxN, soit par une succession de deux convolutions par des filtres de tailles $Nx1$.

En effet, si F represente la transformée de fourrier. La convolution est :

$$I_F = I_I \otimes G$$

d'ou :

$$F(I_F) = F(I_I) \cdot F(G)$$

$$= F(I_I).1/\left(\sqrt{2\pi}\,\sigma\right) \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) e^{-j(w_x x + w_y y)}\,dx\,dy$$

$$= F(I_I).1/\left(\sqrt{2\pi}\,\sigma\right) \int_{-\infty}^{+\infty} \exp\left(-\frac{x^2}{2\sigma^2}\right) e^{-jw_x x}\,dx \int_{-\infty}^{+\infty} \exp\left(-\frac{y^2}{2\sigma^2}\right) e^{-jw_y y}\,dy$$

$$= F(I_I).F(G_x).F(G_y).$$

Il vient :

$$I_F = I_I \otimes G_X \otimes G_Y$$

Nous voyons donc que l'on obtient un résultat identique en convoluant l'image I_I successivement par deux filtres de taille Nx1. Le premier filtre convolue l'image par une gaussienne à une dimension suivant X, le résultat obtenu est convolue suivant Y par un filtre dont les coefficients sont identiques au premier afin de ne pas introduire de distorsions spaciales dans le filtrage. Le processus est résumé sur la Fig.II.7 . On peut toutefois noter que l'ordre des convolutions est indifferent.*[ref 8]*

Le temps de calcul est alors pour chaque filtre de 6σ multiplication est 6σ-1 additions par pixel de l'image, soit un total pour le filtrage de 12σ multiplications et 12σ-2 additions par pixels. Nous voyons donc que dans ce cas le temps de calcul est lineaire en fonction de N, ce qui est un avantage indeniable.

2.2.5.2. Calcul des coefficients du filtre :

On procéde de la maniére suivante :

- Choix de $N=6\sigma+1$, taille du filtre en pixels
- Calcul de $\alpha=(N-1)/6$
- Pour i variant de 0 à N-1 on calcule

$$N(i) = \left[1/\sqrt{2\pi}\,\sigma\right] e^{-\left[\frac{(i-(N-1)/2)^2}{2\sigma^2}\right]}$$

2.2.5.3. Choix de la taille du filtre :

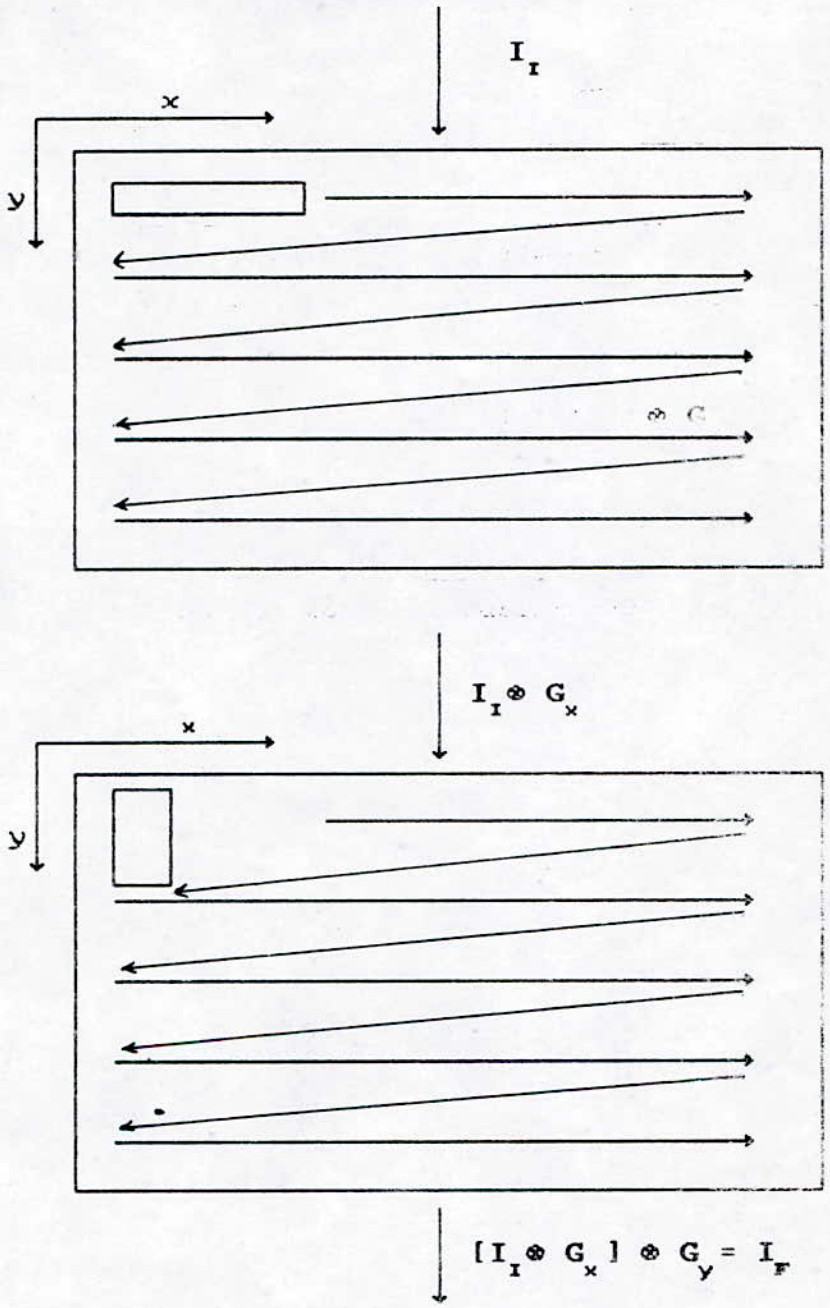


Fig II.7 : *Principe de la convolution par une succession de deux filtres unidimentionels.*

Nous avons vu que, plus la taille du filtre d'un filtre est importante plus l'image filtrée est floue, donc plus puissants sont ses effets (Voir Démo.II.3). Lorsqu'un fort filtrage est necessaire, il faut donc choisir une valeur elevée de N. Or considerons la transformée de Fourrier du filtre, soit *[ref 8]* :

$$g(\omega) = F[G(x,\sigma)] = \exp(-\frac{\sigma^2\omega^2}{2})$$

et considerons un ensemble G_i de filtres gaussiens :

$$G_i(x,\sigma_i) = 1/\left(\sqrt{2\pi}\ \sigma_i\right) \exp\left(-\frac{x^2}{2\sigma_i^2}\right)$$

La TF du produit de convolution est :

$$F\left[\prod_{i=1}^{N} G_i(x,\sigma_i)\right] = \prod_{i=1}^{N}\left[e^{\left(-\frac{\sigma_i^2\omega^2}{2}\right)}\right] = e^{\left(-\frac{\omega^2}{2}\sum_{i=1}^{N}\sigma_i^2\right)}$$

en posant $\sigma^2 = \sum_{i=1}^{N}\sigma_i^2$, l'expression précédente s'écrit :

$$F\left[\prod_{i=1}^{N} G_i(x,\sigma_i)\right] = F\left[G(x,\sigma)\right] = e^{-\frac{\omega^2}{2}\sigma^2}$$

En appliquant la transformée de Fourrier inversse, il vient :

$$\prod_{i=1}^{N} G_i(x,\sigma_i) = G(x,\sigma)$$

Donc un filtre Gaussien ayant une fenêtre de grande taille peut-être décomposé en une cascade de filtres gaussiens de taille plus petite.

2.2.5.4. Organigramme du filtre :

Variables :

TailleFiltre.....: Taille du filtre.

DemiTaille.....: Demi taiile du filtre.

Sigma.....: Variance.

Filtre.....: Vecteur des coefficients du filtre.

Somme.....: Somme des coefficients.

Ima.....: Image initiale qui recevera les résultats.

ImaTemp.....: Image intermediaire.

Le shéma de l'organigramme est sur la Fig II.8.

2.3. Le filtrage non lineaire :

Ce type de filtrage, s'oppose au précédent dans sa dénomination car il n'est pas le resultat d'une combinaison lineaire de pixels.

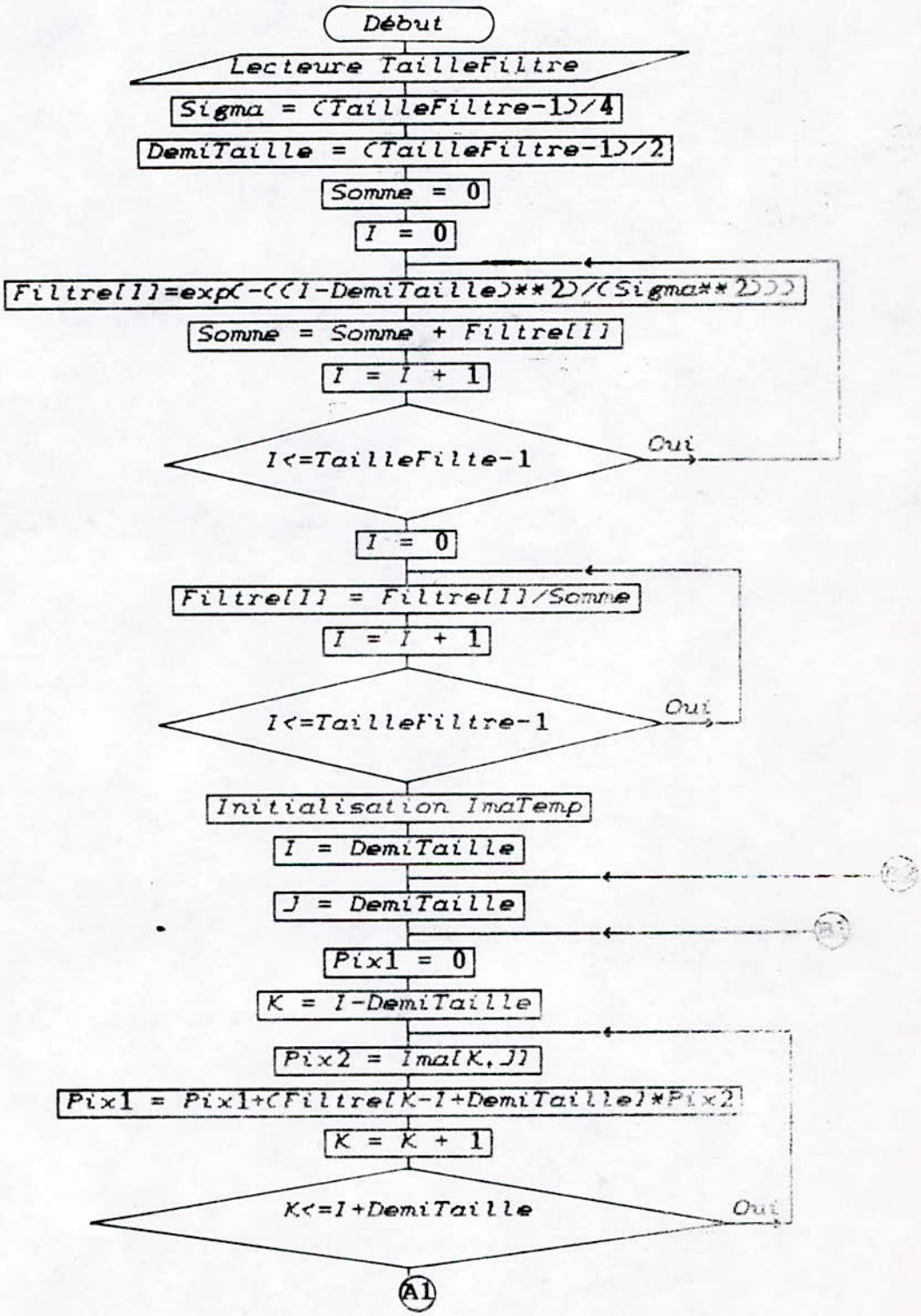
2.3.1. Le filtre median :

L'exemple le plus classique du filtrage non lineaire est le *filtre median de Tuckey*. Son principe est illustré sur la figure II.9 et il peut être decomposé selon les etapes suivantes :

Pour chaque pixel de l'image :

- On classe les pixels voisins du pixel courant(compris dans la fenêtre) par valeurs croissantes;
- On prend la valeur médiane des pixels classés et on l'affecte au pixel courant.

2.3.2. Taille et forme du filtre median :


Début
Lecteure TailleFiltre
Sigma = (TailleFiltre-1)/4
DemiTaille = (TailleFiltre-1)/2
Somme = 0
I = 0
Filtre[I]=exp(-((I-DemiTaille)**2)/(Sigma**2)))
Somme = Somme + Filtre[I]
I = I + 1
I<=TailleFilte-1
Oui
I = 0
Filtre[I] = Filtre[I]/Somme
I = I + 1
I<=TailleFiltre-1
Oui
Initialisation ImaTemp
I = DemiTaille
J = DemiTaille
Pix1 = 0
K = I-DemiTaille
Pix2 = Ima[K,J]
Pix1 = Pix1+(Filtre[K-I+DemiTaille]*Pix2
K = K + 1
K<=I+DemiTaille
Oui
A1

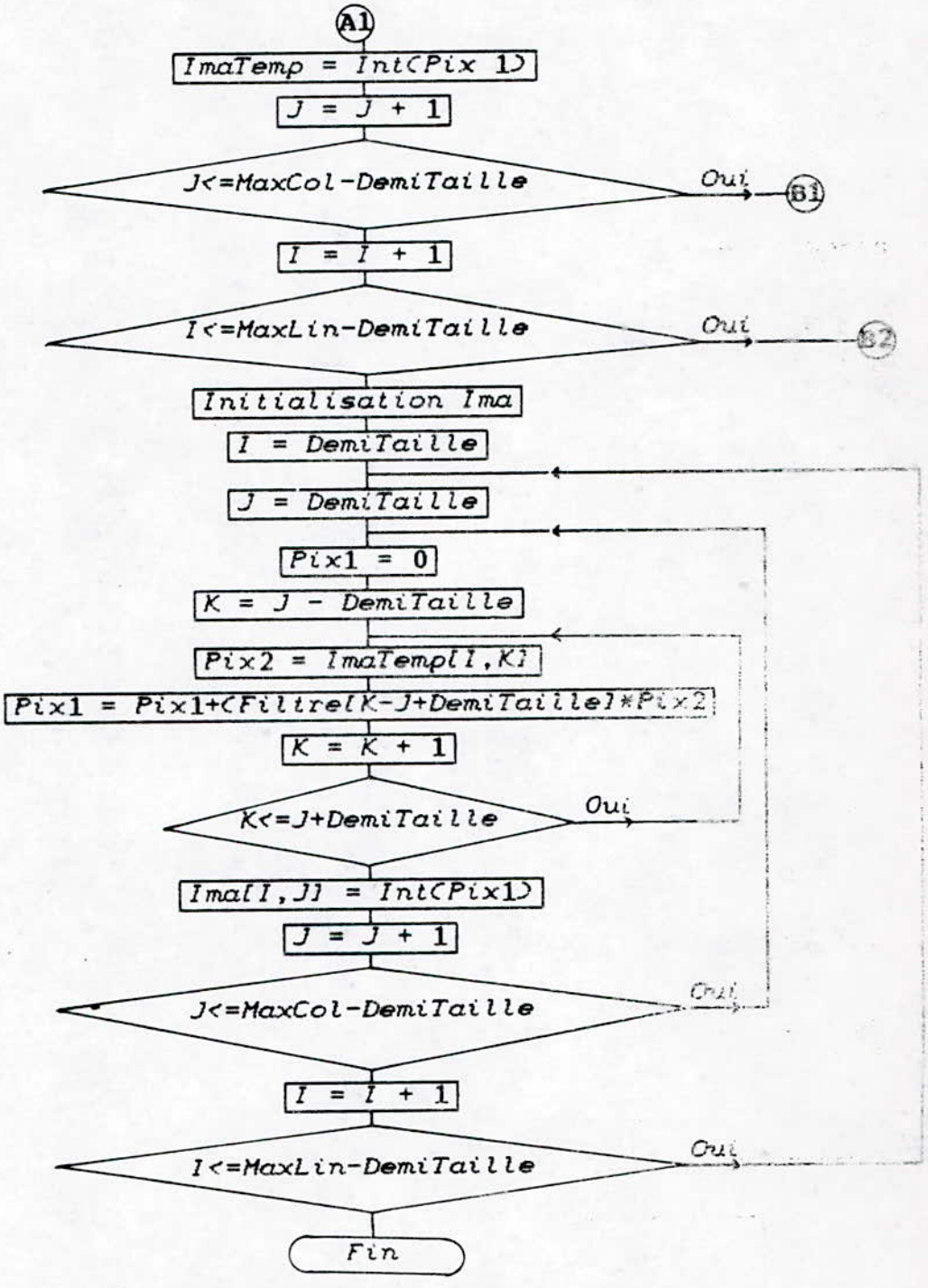


Fig II.8 : *Organigramme du filtre de GAUSS.*

Plus la taille du filtre est grande plus le filtrage peut paraitre efficace, mais plus il déforme l'image sans pour autant adoucir son contraste. En fait, le filtre median est efficace lorsque l'image est dégradée par une source de bruit de type impultionnel (neige), donc lorsqu'on assiste à des variations brusques de pixels isolés. En contre partie ce filtre introduit une petite perte de résolution.(Voir Démo.III.4)

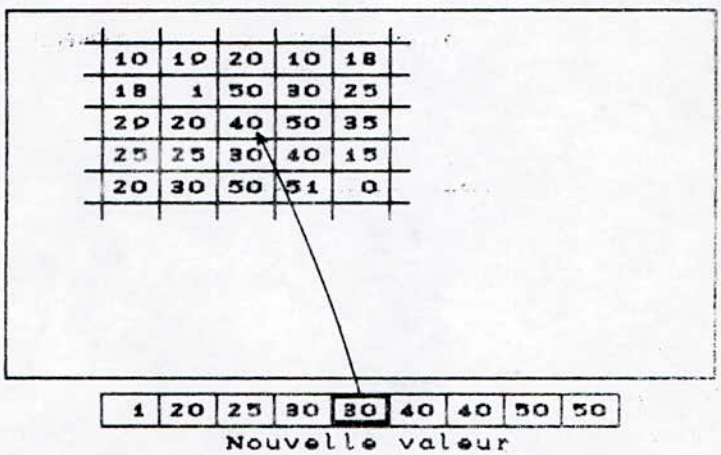


Fig II.9. : *Principe de fonctionnement du filtre median.*

D'autres masques en forme de croix, a été préféré un masque carré classique, engendrant moins de déformations, tout en étant aussi efficaces vis à vis des perturbations agissant sur des pixels isolés.(voir Fig II.10).

2.3.3. Organigramme du filtre median :

Variables :

ImaIn : Image initiale.

ImaOut : Image finale.

Tri : Table des tris.

EndTri : Variable booléenne.

L'organigramme est sur la Fig II.11.

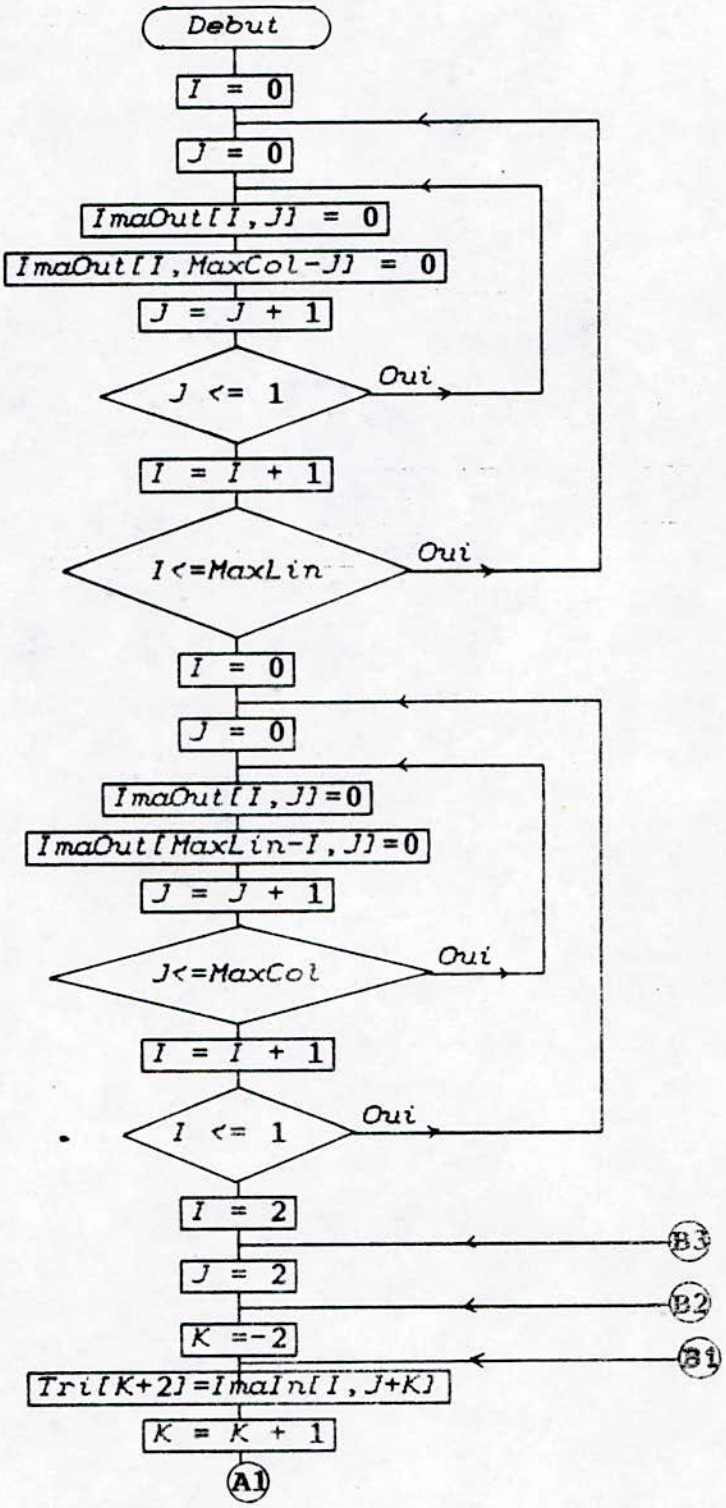
.a. .b.

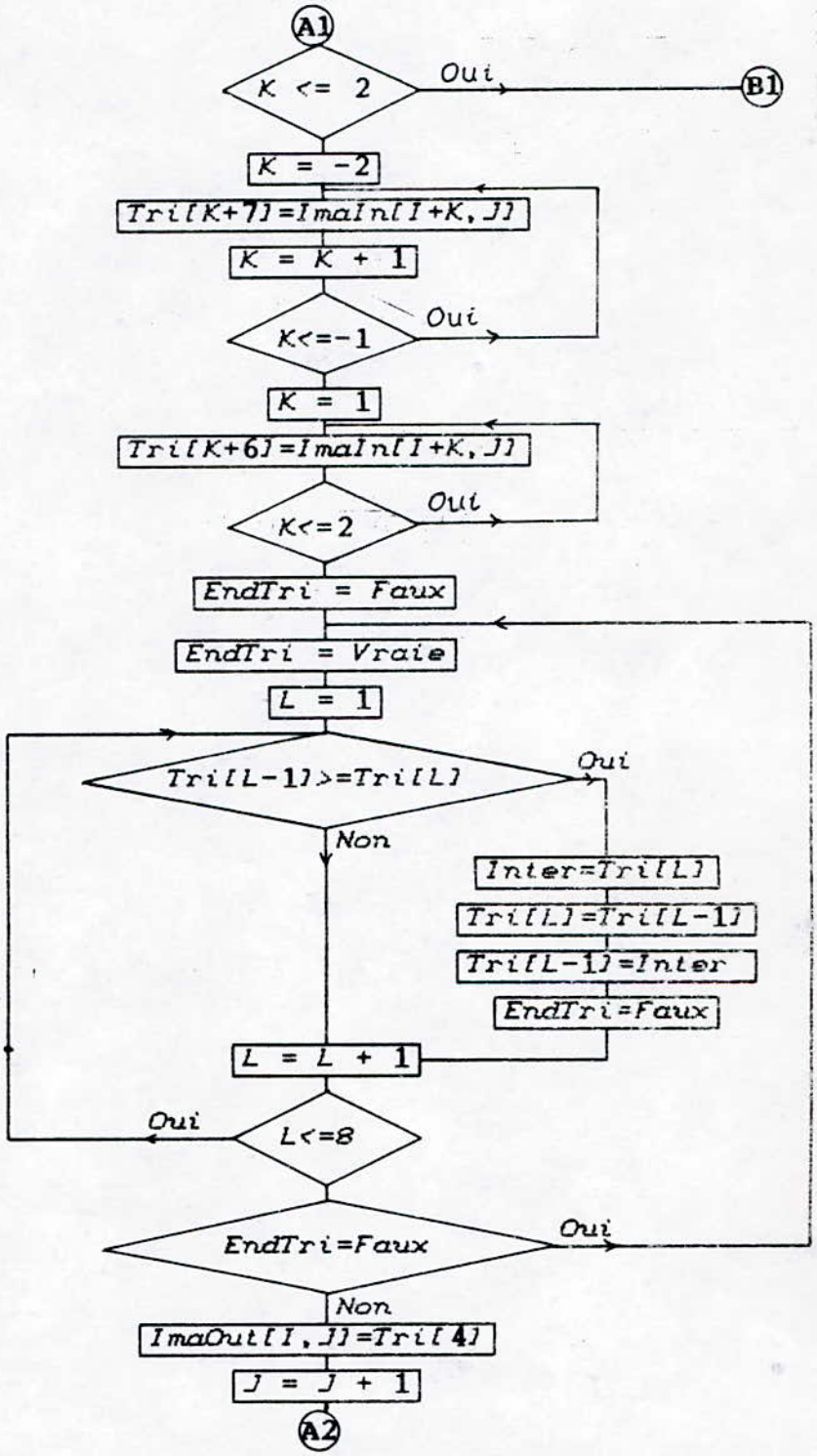
Fig II.10. : *Masques utilisées par le filtre median. a: Masque en croix, b: Masque carré.*

2.4 CONCLUSION SUR LE FILTRAGE :

Cet important paragraphe nous a permis de voir qu'il existe deux techniques pour filtrer une image : la première dans le domaine fréquenciel avec les filtres passe-bas et passe-haut qui font appel aux FFT; la deuxiéme dans le domaine spacial avec les techniques de masquage qui s'appuient sur les propriétés de la convolution d'une image par une fenêtre.

Pour finir, notons que pour les images bruitées binaires c'est-à-dire à gamme de gris reduite au noir ou blanc, il existe des techniques spécifiques dites *Opérateurs morphologiques* que nous developperons par la suite.


Debut
I = 0
J = 0
ImaOut[I,J] = 0
ImaOut[I,MaxCol-J] = 0
J = J + 1
J <= 1
Oui
I = I + 1
I<=MaxLin
Oui
I = 0
J = 0
ImaOut[I,J]=0
ImaOut[MaxLin-I,J]=0
J = J + 1
J<=MaxCol
Oui
I = I + 1
I <= 1
Oui
I = 2
J = 2
K =-2
Tri[K+2]=Imain[I,J+K]
K = K + 1
A1
B3
B2
B1


A1
K <= 2
Oui
B1
K = -2
Tri[K+7]=Imain[I+K,J]
K = K + 1
K<=-1
Oui
K = 1
Tri[K+6]=Imain[I+K,J]
K<=2
Oui
EndTri = Faux
EndTri = Vraie
L = 1
Tri[L-1]>=Tri[L]
Oui
Non
Inter=Tri[L]
Tri[L]=Tri[L-1]
Tri[L-1]=Inter
EndTri=Faux
L = L + 1
Oui
L<=8
EndTri=Faux
Oui
Non
ImaOut[I,J]=Tri[4]
J = J + 1
A2

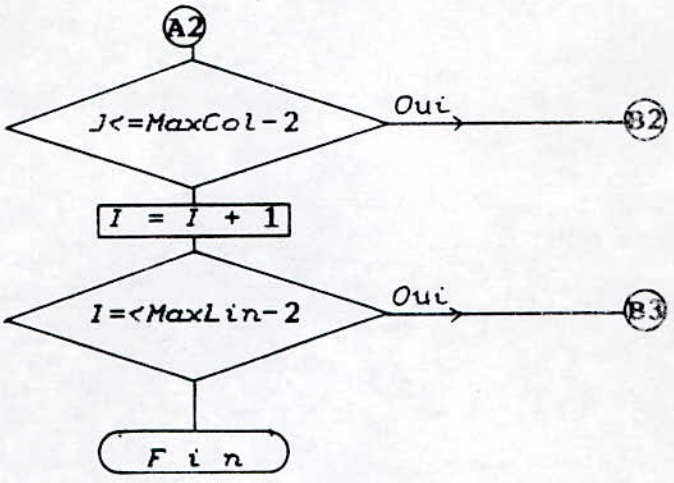
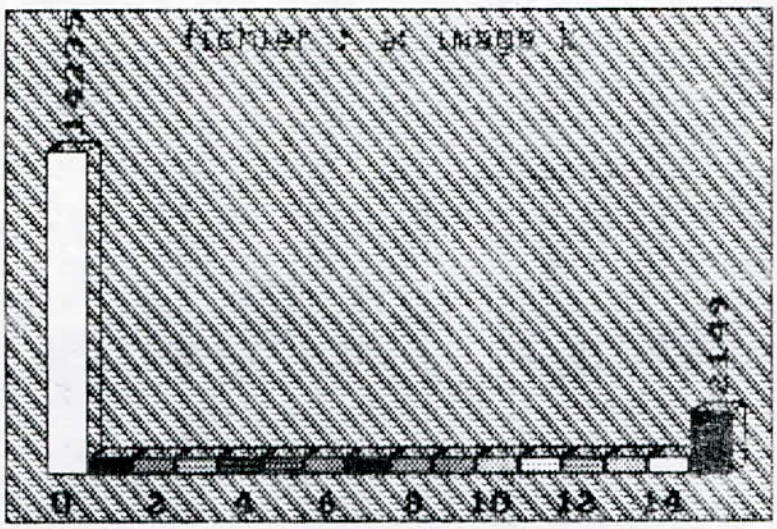
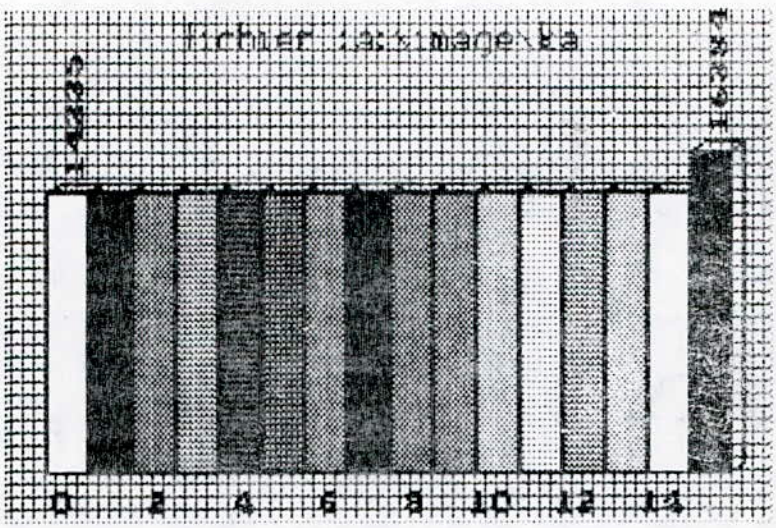


Fig II.11 : *Organigramme du filtre median.*

a

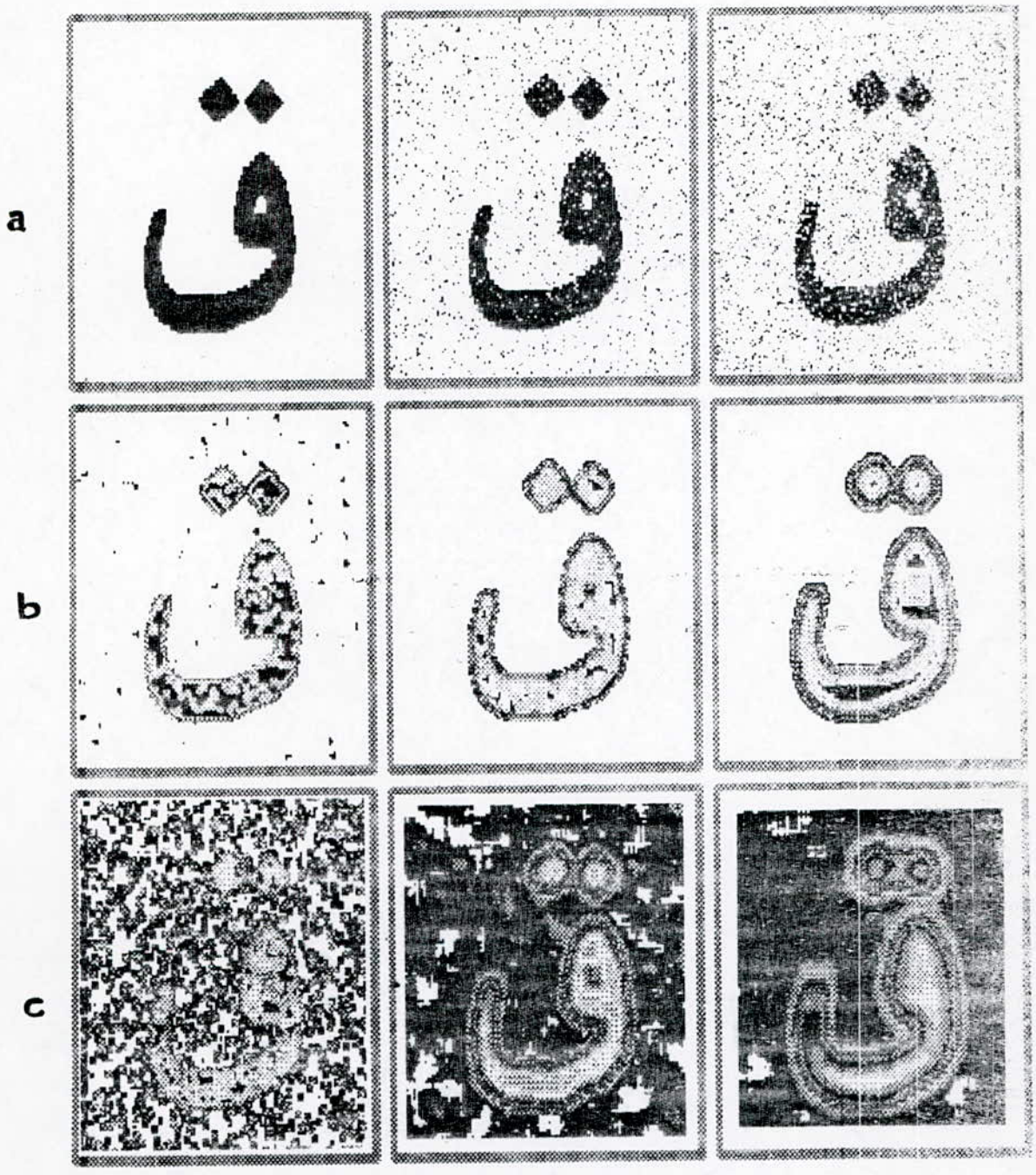


b

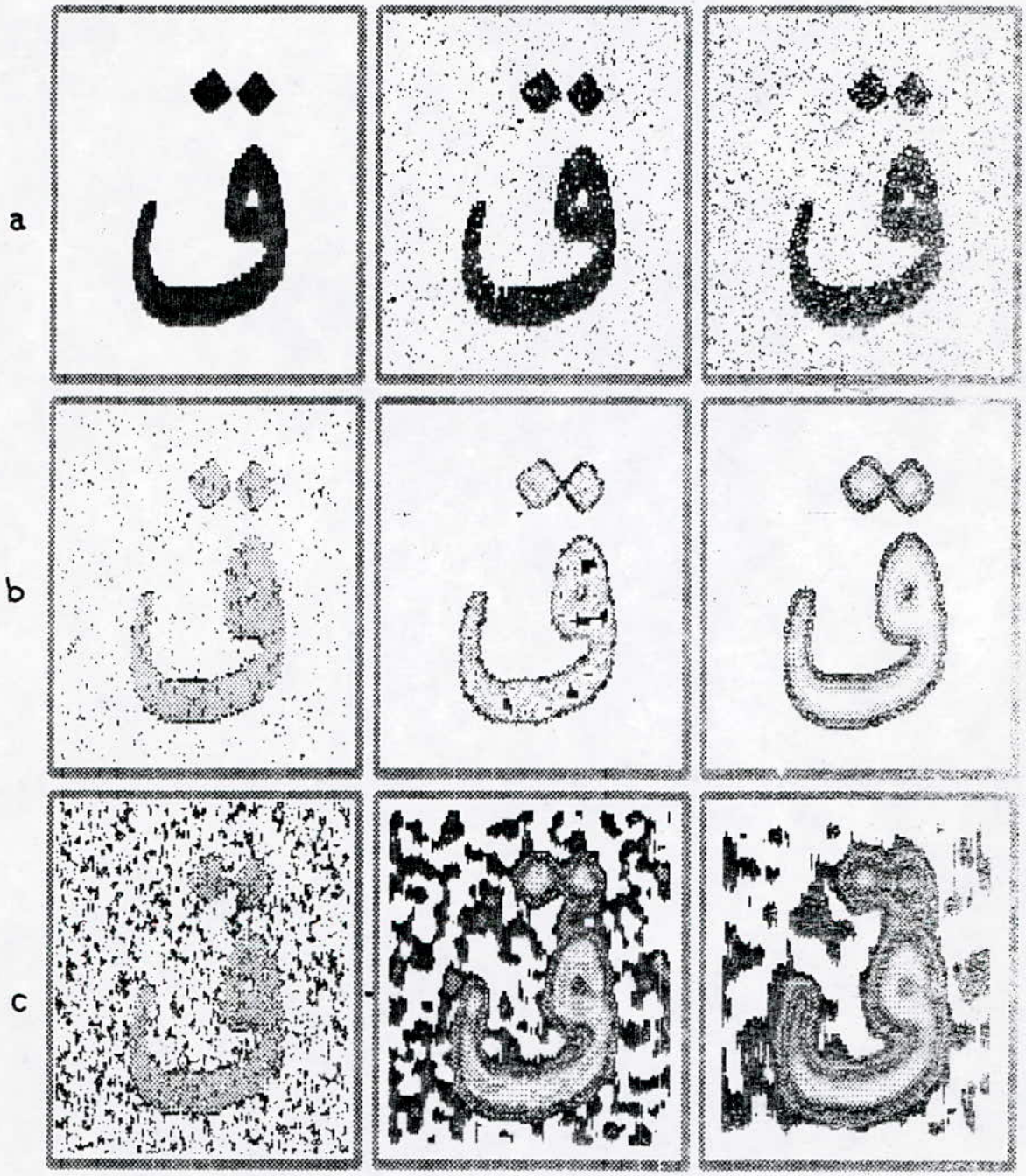


c

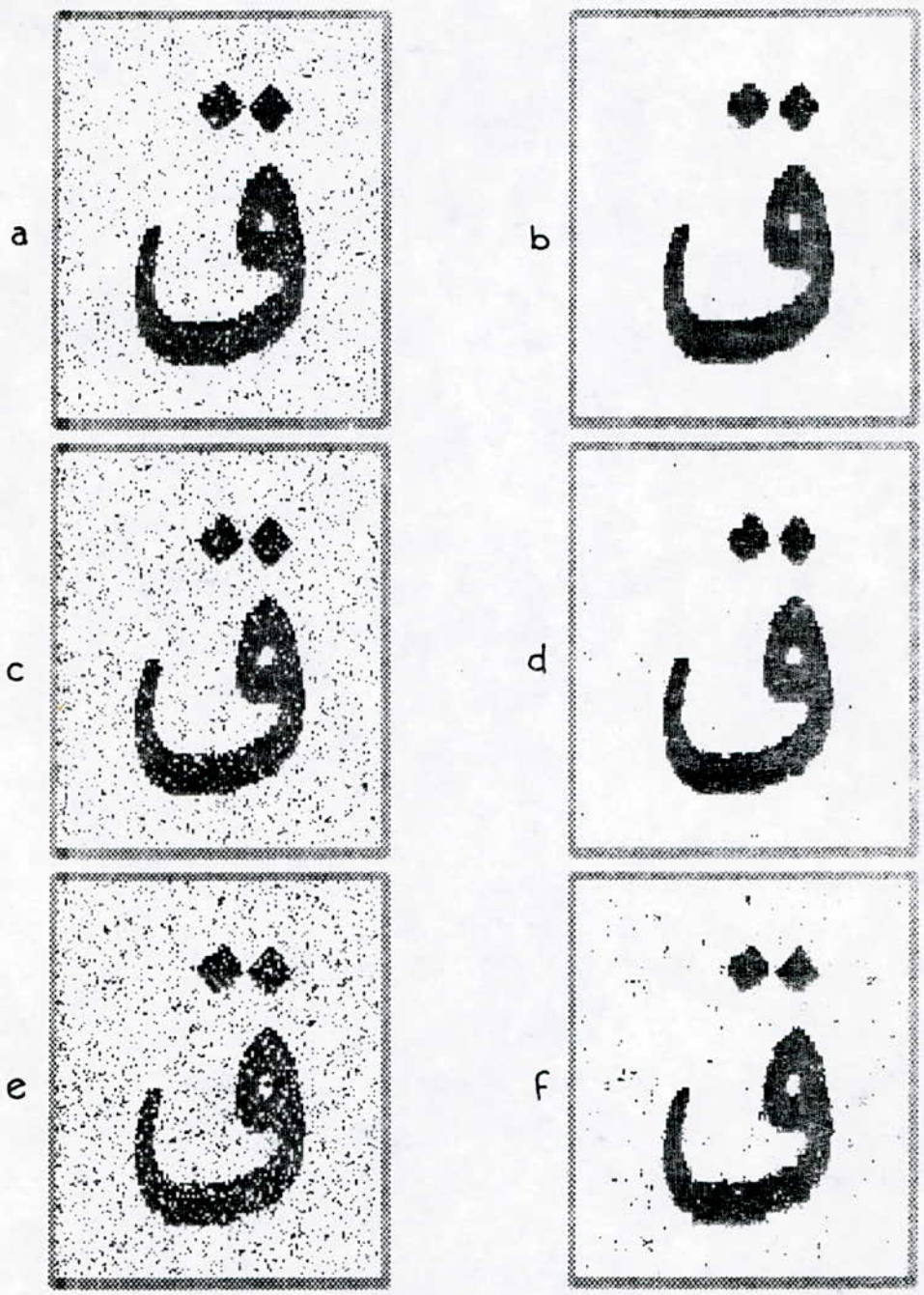
Demo II.1. *Histogramme d'une image; (a) : Image binaire. (b) : Histogramme Simple; (c) : Histogramme cumule.*



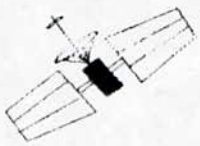
Demo II.2. *Filtrage moyen. (a): image originale; 1ère image bruitée (d=10,n=5); 2ème image bruitée(d=5,n=15). (b): 1ère image filtrée avec les masques de taille 3,5 et 9. (c): 2ème image filtrée avec les masques de taille 3,9 et 15.*



Demo II.3. *Filtre de* **Gauss**. *(a): image originale; 1^ere image bruitee (d=10,n=5); 2^eme image bruitee(d=5,n=15). (b): 1^ere image filtree avec les masques de taille 3,5 et 9. (c): 2^eme image filtree avec les masques de taille 3,9 et 15.*



Demo II.4. *Filtre Median. (a): Image bruitee (d=10,n=5). (b): Image filtree. (c): Image bruitee(d=8,n=10). (d): Image filtree. (e): Image bruitee (d=5,n=15). (f): Image filtree.*



CHAPITRE III

SEGMENTATION
EXTRACTION DES CONTOURS

La segmentation est l'opération qui consiste à subdiviser une scène réelle en ses parties constituantes ou objets. Ces régions peuvent être caractérisées par leurs frontières et c'est le cas de la segmentation par extraction de contours ou bien être directement caractérisées par les pixels qui les composent, il s'agit alors de la segmentation en régions homogènes.

Il est évident que ces deux approches de la segmentation sont duales. Cependant l'information qu'elles mettent en évidence est différente. Les contours possédent l'essentiel des caractèristiques de forme de la région; la segmentation en regions homogénes fait intervenir les caractèristiques non gèometriques liées au critère de segmentation, donc au contenu de la region plus qu'à sa forme.

Ces dernières considérations nous amènent à utiliser une première approche de la segmentation vu qu'on a à faire à un probléme de reconnaissance de forme.

1.EXTRACTION DE CONTOUR :

Le but de l'analyse d'image est d'obtenir une description synthètique des divers éléments qui la constituent à partir de la masse enorme d'information qu'elle contient à l'état brut. L'un des processus fondamentaux dans la chaine de reconnaissance visuelle consiste à diminuer cette quantité d'information en ne gardant que les points essentiels de l'image. Les points du contour constituent en ce lien une approche rationnelle. Des études ont en effet montré que

l'homme est capable de reconnaitre un objet par simple observation de ses contours.

La notion de contour dans une image est fortement liée à ce que l'on désire voir dans celle-ci. Il est donc nécéssaire dans l'idéal d'avoir une information à priori sur le contexte telle que taille des objets, contraste moyen de ceux-ci avec le fond etc .... Ceci n'est autre que la définition globale de l'œil humain. Actuellement les systèmes de vision industriels ne fonctionnent pas encore sous ce principe ideal mais simplement en chaine directes, les paramétres et le choix des opérateurs étant determinés en fonction de l'application.

1.1.Definition du contour :

Les considerations précèdentes nous amènent à devoir fixer une définition mathèmatique d'un contour, independamment de tout critère subjectif.

On definit ainsi un contour comme étant une brusque variation de niveau de gris dans l'image (Voir Fig.III.1). On peut caracteriser cette variation par son amplitude (Paramétre h) et par sa pente (Paramétre ρ). Cette definition détermine donc toute variation de niveau de gris, aussi faible soit-elle, comme etant un point de contour de plus ou moins grande intensité. Le choix des contours que l'on désire éliminer ou garder peut alors être déterminé en seuillant ceux-ci.

1.2.Processus d'extraction :

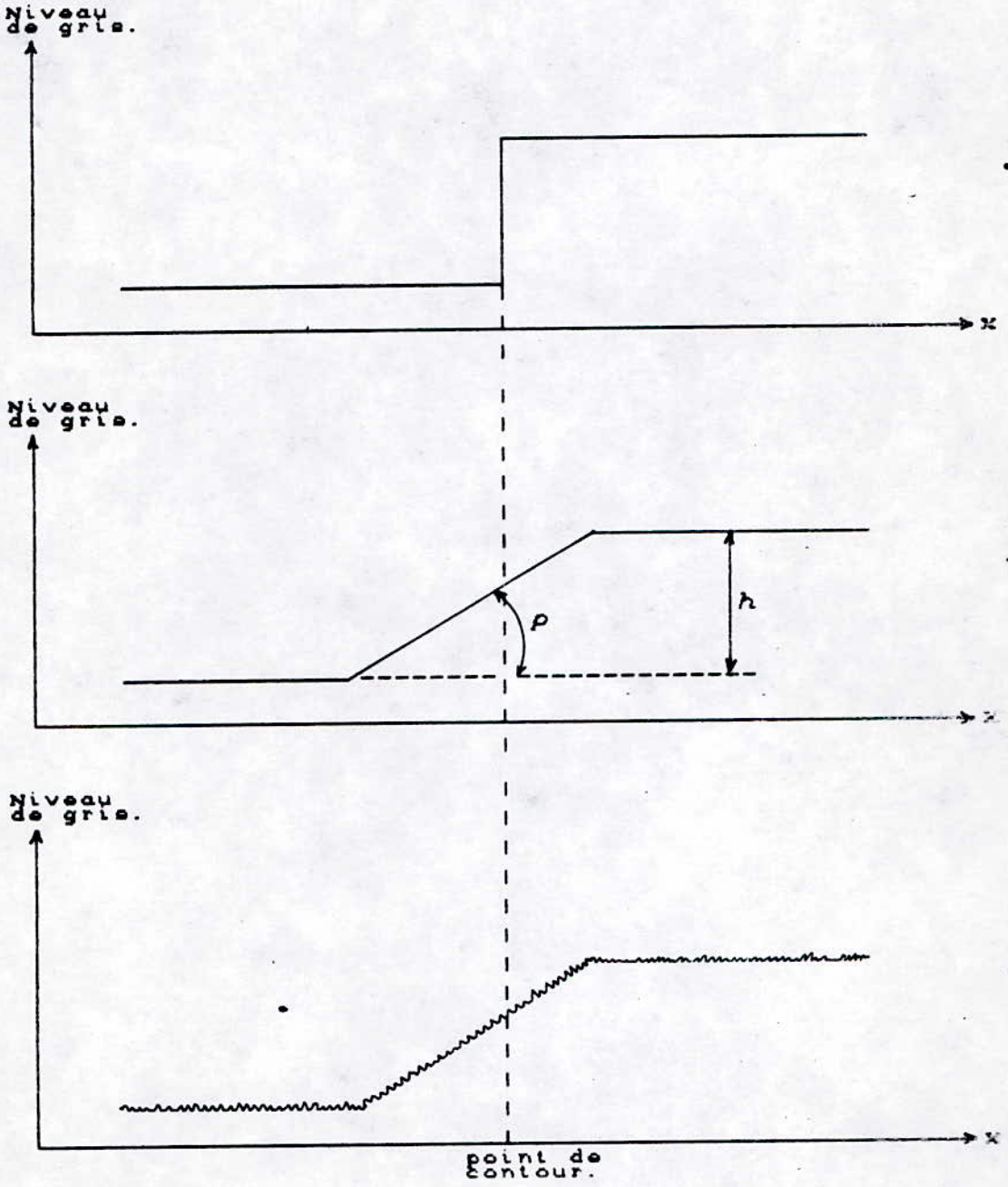


Fig III.1 : *Exemples de contours.*

(a): Contour ideal, (b): Rampe, (c): Contour réel.

Le processus d'extraction de contours peut être décomposé en plusieurs étapes distinctes qui peuvent être décrites comme suit :

1.1.2.Mise en evidence des contours :

Elle s'obtient par une differentiation de l'image. Elle peut s'appliquer à des images binaires ou non. Dans le cas d'images initiales binaires, le résultat peut être directement binaire lui aussi suivant le type d'opérateur de dérivation employé. Ce qui permet de sauter la seconde étape du processus d'extraction. Signalons de plus que dans le cas d'images binaires, il existe aussi des extracteurs de contours dits *topologiques* qui procédent à un suivi du contour sans effectuer de dérivation. Ils fournissent directement un contour binaire.

1.2.2.Binarisation des contours :

Lorsque le contour présente plusieurs niveaux de gris on peut avoir recours à la binarisation. Cette procédure est decrite dans le prochain chapitre.

2.OPERATEURS DE DIFFERENTIATION :

Nous venons de voir qu'un premier traitement dans le processus, l'extraction des contours consiste à mettre ceux-ci en évidence. Ceci est réalisé en dérivant l'image une fois ce qui permet d'obtenir un *Gradient*, ou bien en derivant l'image deux fois et on obtient alors un *Laplacient*. La Fig.III.2 montre les effets de la derivation en présence d'un contour.

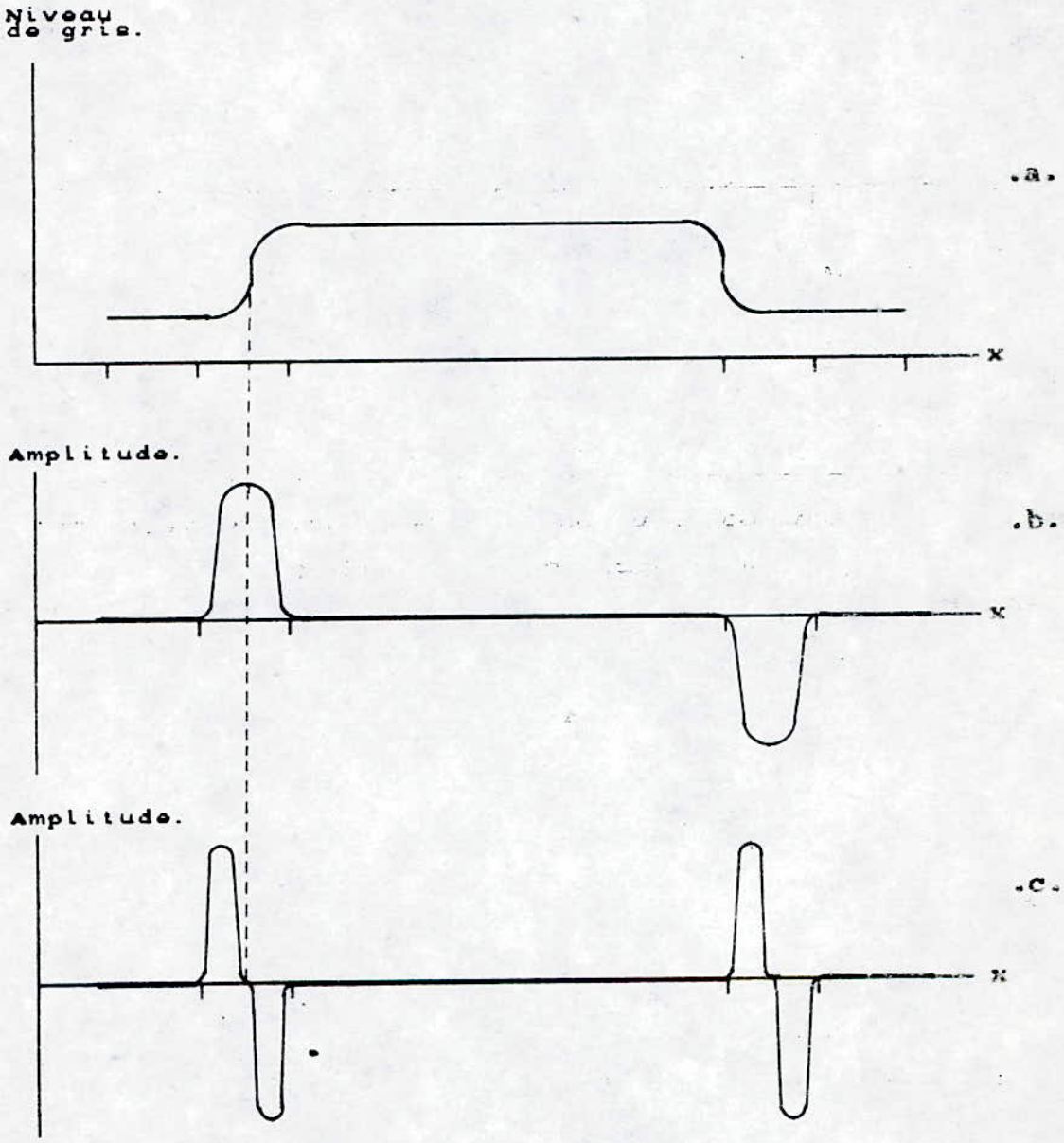


Fig III.2. *Derivation en présence d'un contour*
(a): Contour, (b): Gradient, (c): Laplacien.

Dans de nombreux ouvrages les opérateurs de différenciation sont considèrés comme faisant partie des prétraitements.Nous les avons néamoins séparés des autre types de filtrage du fait de leurs applications spécifiques à l'extraction des contours.

2.1.Le Gradient :

Les opérateurs de dérivation sont nombreux. Parmi ceux-ci, le premier développé fût celui de *Robert* en 1965 et n'est autre qu'une application directe de la formule d'une dérivée. Il se présente sous la forme suivante pour chaque pixel $I(x,y)$ d'une image I.

Les dérivées en x et y sont :

$$A_x = I(x,y+1) - I(x,y)$$
$$A_y = I(x+1,y) - I(x,y)$$

ce qui revient à convoluer l'image avec les masques :

$$H_x = \begin{pmatrix} -1 & +1 \end{pmatrix} \qquad H_y = \begin{pmatrix} -1 \\ +1 \end{pmatrix}$$

L'amplitude du gradient est alors :

$$A(x,y) = \sqrt{A_x^2 + A_y^2}$$

La direction du gradient est donnée par :

$$D(x,y) = \mathrm{Arctan}\left(A_x / A_y\right)$$

Pour des raisons de rapidité de calcul, le calcul de l'amplitude du gradient a été modifié. Nous avions précédemment défini celle-ci comme étant :

$$A_o = \sqrt{A_x^2 + A_y^2}$$

La nouvelle expression est donnée par :

$$A_1 = \mathrm{Max}\left(|A_x|, |A_y|\right)$$

Notons que celle-ci introduit des distortions dans le calcul de l'amplitude car elle dépend de la direction du gradient et non de sa seule amplitude. En effet, lorsque A_x ou A_y sont nul, nous avons $A_1 = A_o$, mais par exemple pour le cas simple $A_x = A_y$, nous avons $A_1 = A_x$ alors que $A_o = A_x\sqrt{2}$. Les raisons de ces distortions sont simples : les masques des opérateurs constituent une approximation discrète de la dérivée réelle et de ce fait la norme à appliquer pour le calcul de l'amplitude est dépendante des valeurs des composantes A_x et A_y.

2.1.1.Masque de Roberts :

Avec l'opérateur de *Roberts*, la detection de contour, (voir **Démo.III.1.b**), s'effectue en convoluant l'image initiale,avec les masques suivants :

$$H_x = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{pour le gradient suivant X}$$

$$H_y = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{pour le gradient suivant Y}$$

L'inconvenient majeur de cet opérateur, est sa sensibilité au bruit.(Voir Démo.III.2)

Organigramme :

L'organigramme de ce masque est sur la Fig.III.3, les variables utilisées sont :

ImaIn :Image initiale.
ImaOut :Image finale.
GradX :Gradient suivant X.
GradY :Gradient suivant Y.

2.1.2.Masque de Sobel :

Contrairement à l'opérateur de Roberts, cet opérateur présente l'avantage d'être moins sensible au bruit.(Voir Démo.III.1.d)

$$H_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad H_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}.$$

En effet, tout processus de dérivation d'un signal tend à accentuer le bruit présent dans le signal. La sensibilité au bruit est diminuée en effectuant une moyenne locale sur le domaine couvert par le masque.(Voir Démo.III.2)

Organigramme :

L'organigramme de cet opérateur est donné dans la Fig.III.4, les variables utilisées sont :

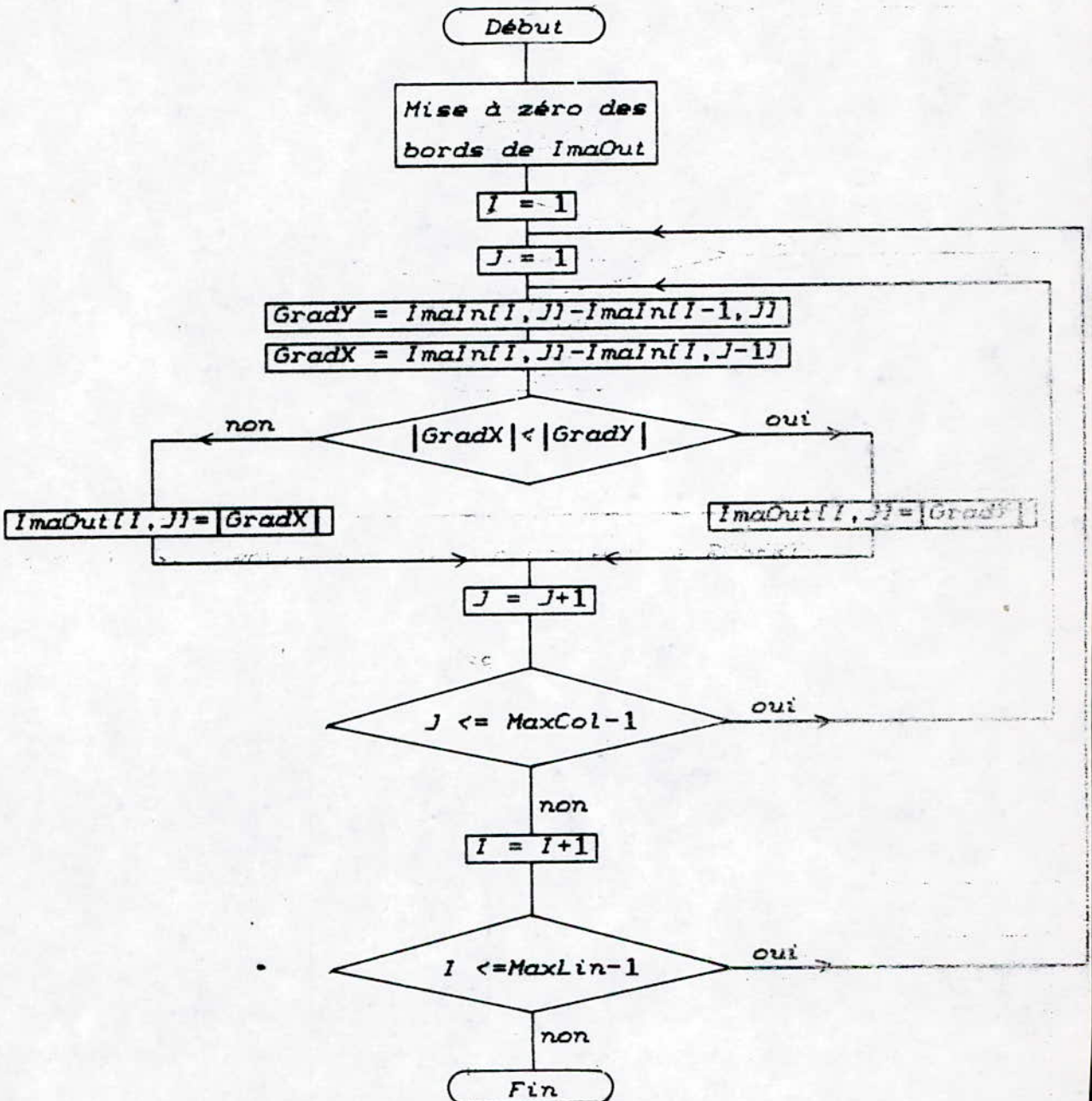


Fig III.3. *Organigramme de l'opérateur de ROBERTS.*

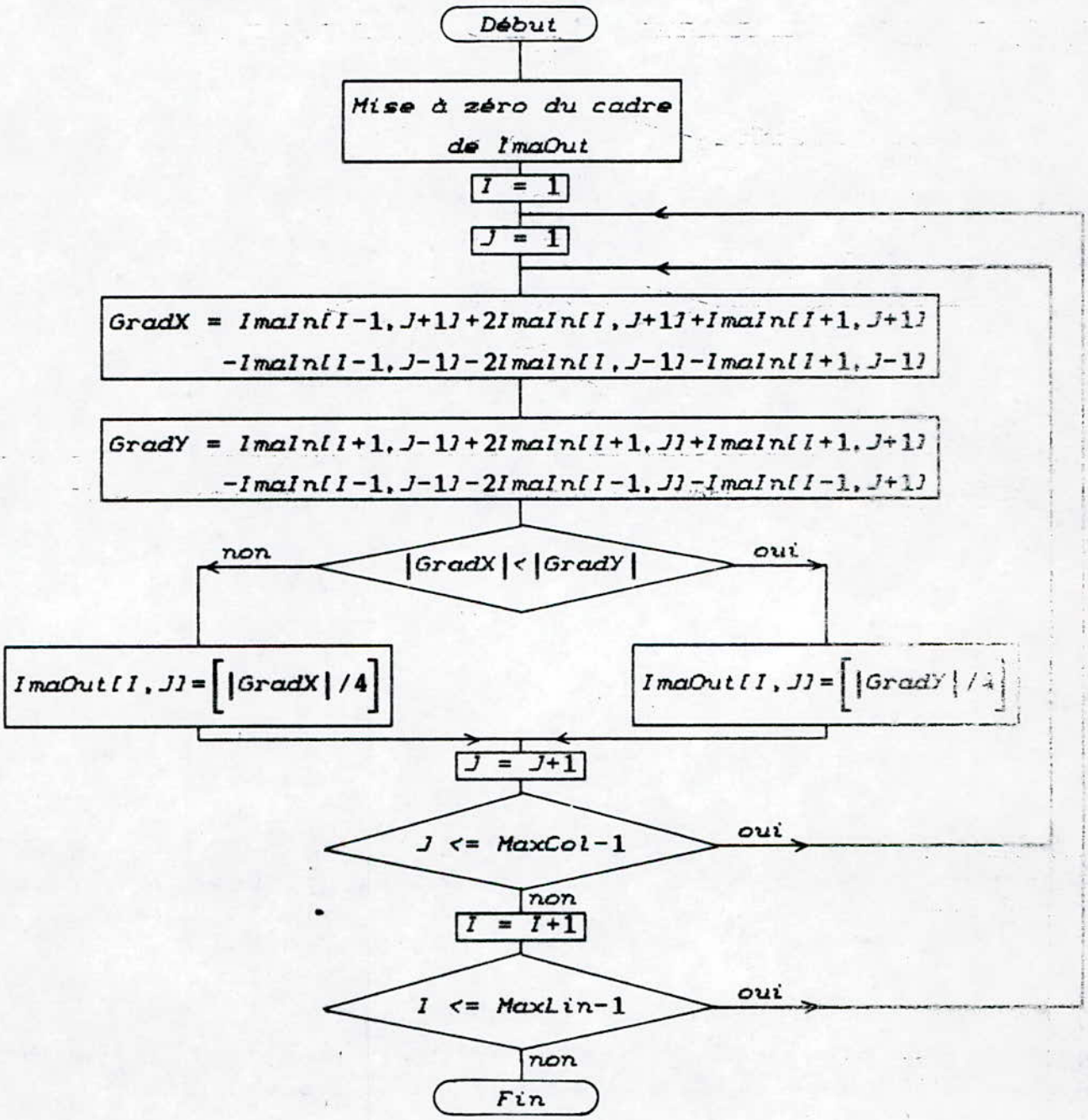


Fig III.4. *Organigramme de l'opérateur de SOBEL.*

ImaOut :Image finale.
GradX :Gradient suivant X.
GradY :Gradient suivant Y.

2.1.3. Masque de Perwitt :

L'opérateur de *Perwitt* agit de la même manière que le précédent, en effectuant la moyenne. La seule différence réside dans les coefficients du masque.

$$H_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad H_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Les résultats donnés par ces deux derniers opérateurs sont satisfaisants en ce qui concerne la sensibilité au bruit, mais présentent l'inconvénient de donner un contour épais. Dans ce cas on peut avoir recours à des méthodes d'amincissement des lignes du contour.

Organigramme :

L'organigramme de cet opérateur est donné dans la Fig.III.5, les variables utilisées sont :
ImaIn :Image initiale.
ImaOut :Image finale.
GradX :Gradient suivant X.
GradY :Gradient suivant Y.

2.2.Le Laplacien :

Le Laplacien d'une image représente sa derivée seconde. Cette dérnière peut s'exprimer de plusieurs manières selon les différences finies utilisées pour approximer les dérivées partielles (centrées, à droite, à gauche). De ce fait, plusieurs masques de laplaciens trés similaires ont été developpés :(Voi Démo.III.1.f)

$$H_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad H_2 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad H_3 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 8 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Comme indiqué sur la Fig.III.2 les points du contour correspondent à un laplacien nul bordé par deux extremums. Le passage par zéro étant unique, le laplacien fournit des contours d'un pixel d'épaisseur. Cependant, le laplacien possède un inconvénient majeur qui est sa grande sensibilité au bruit. En effet, cet opérateur réalise une derivée seconde de l'image et est donc trés instable.(Voir Démo.III.2)

Organigramme :

L'organigramme de cet opérateur est sur la FIG.III.6, les variables utilisées sont :

Ima :Image initiale qui recevra le résultat final.
ImaAux :Matrice des dérivées secondes.
Scale :Facteur d'echelles pour le recadrage.

3.CONCLUSION SUR LA SEGMENTATION :

La méthode de balayage par masque est trés couramment employée, puisque finalement elle englobe les opérateurs de

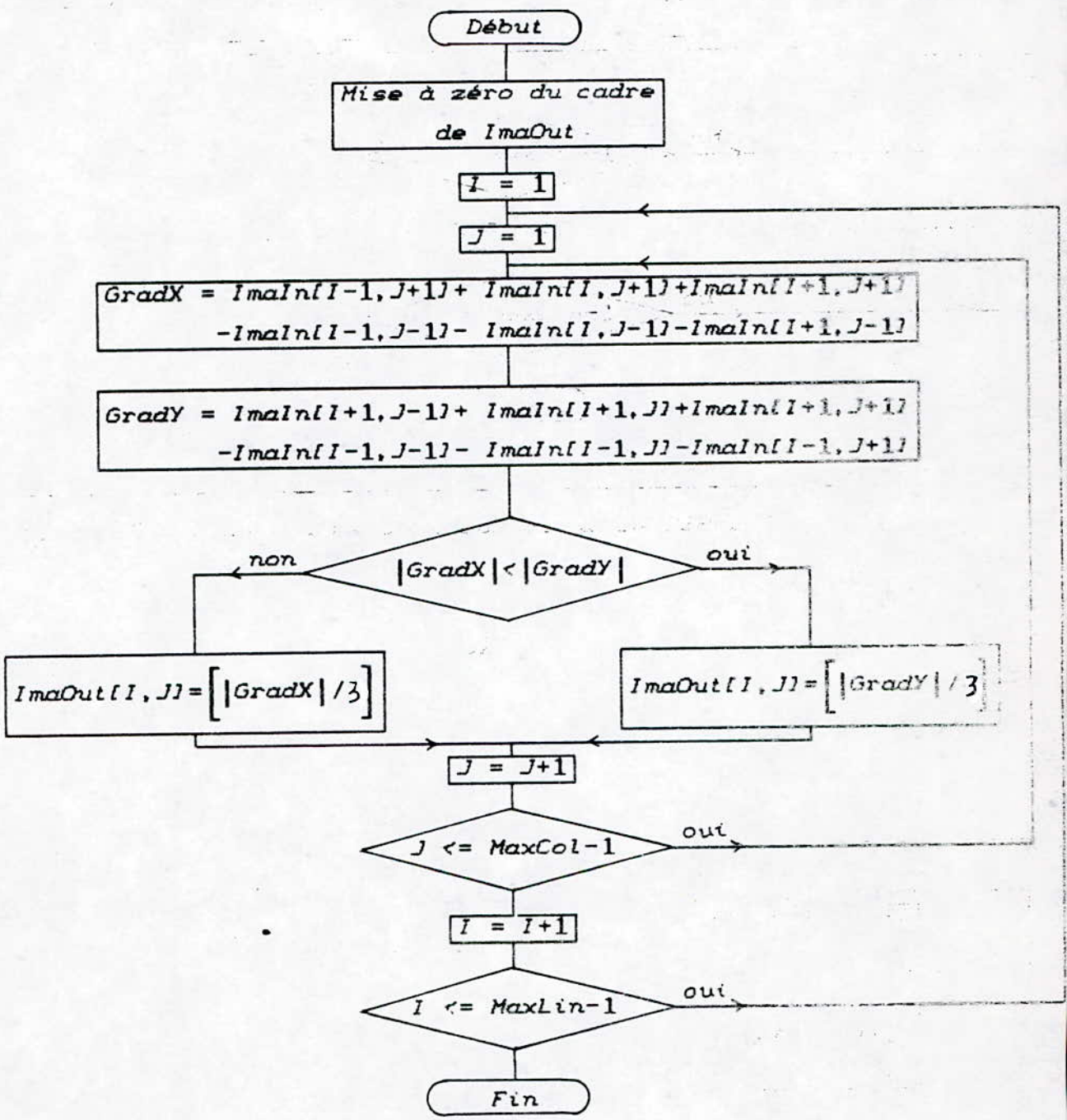
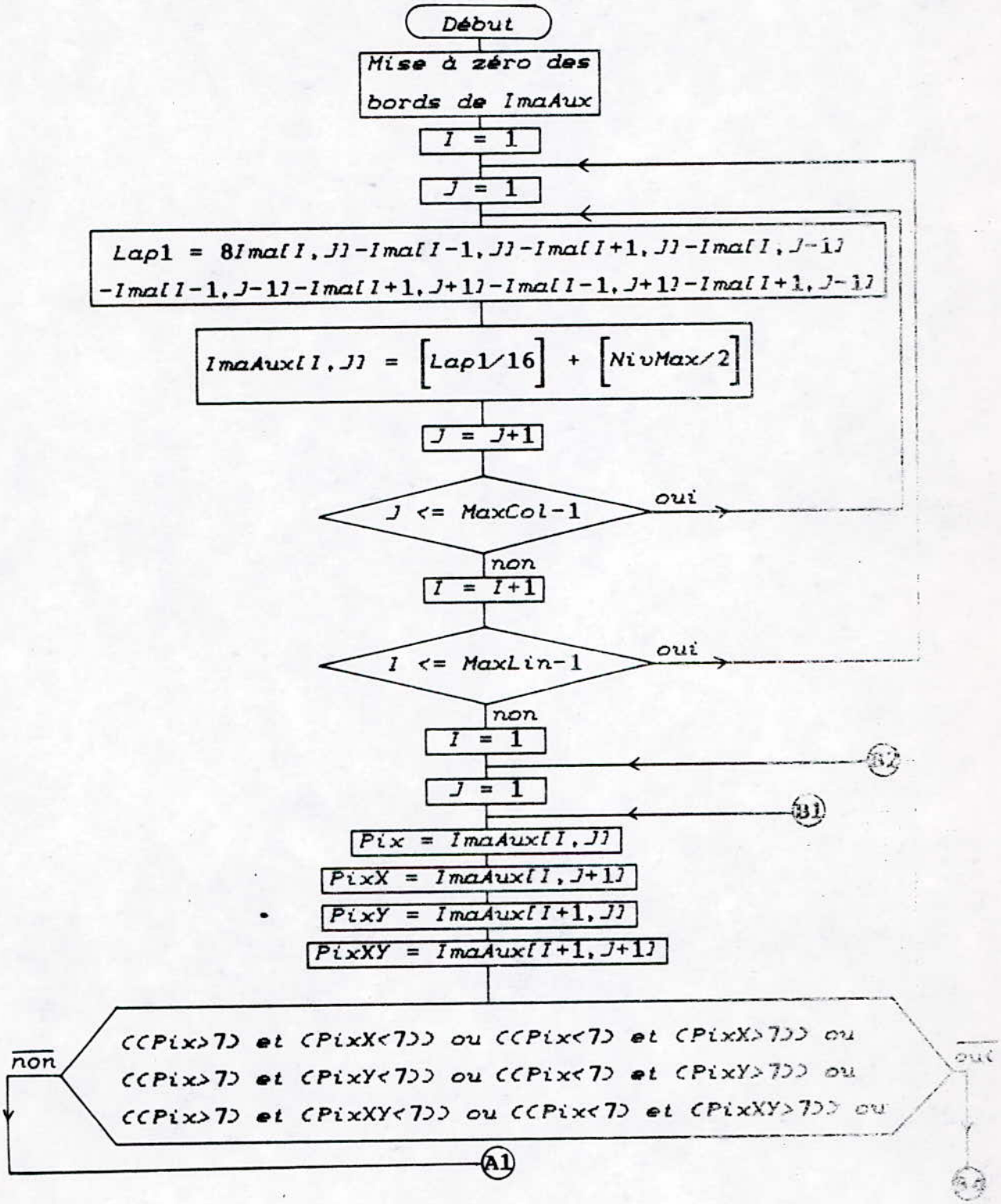
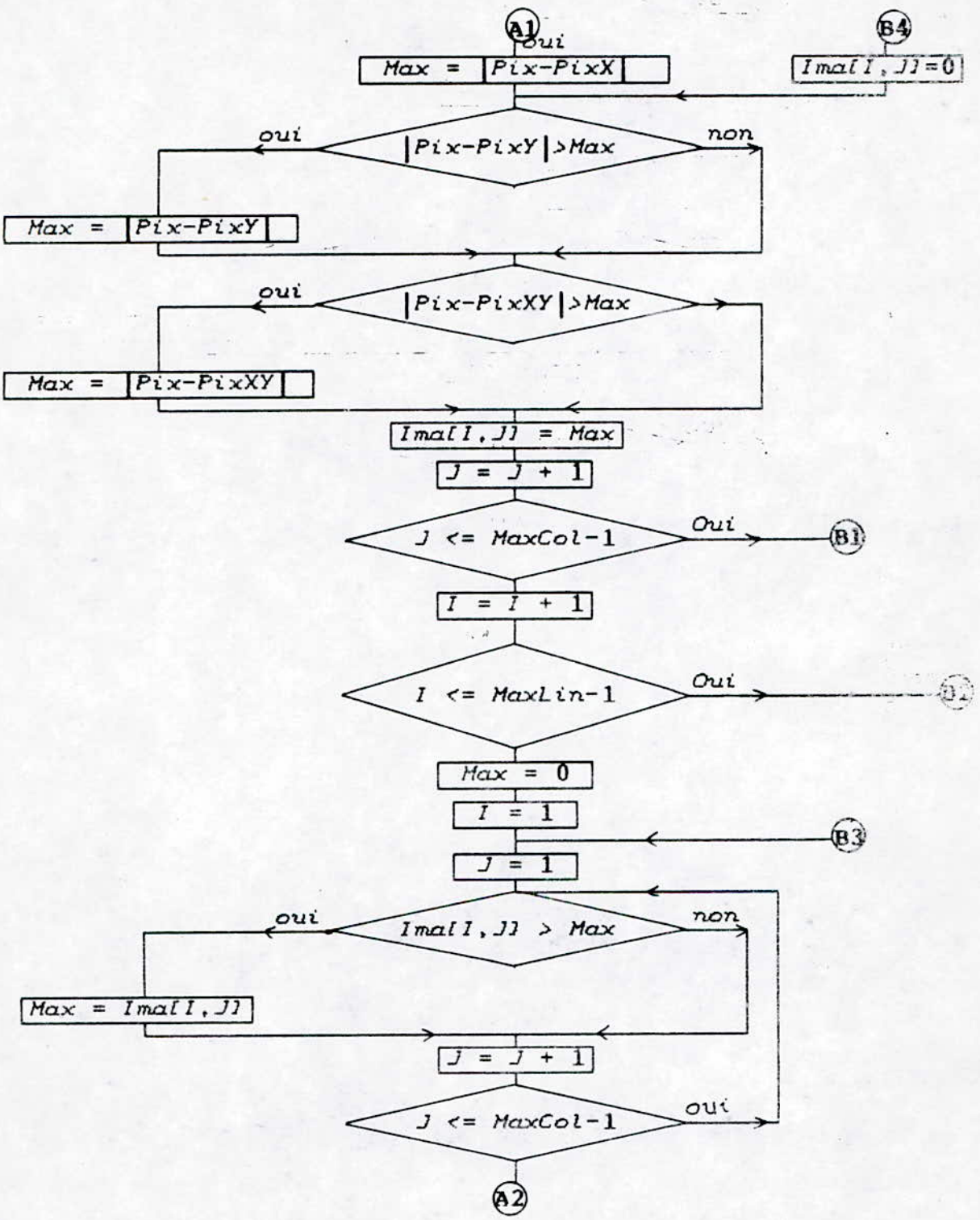


Fig III.5. *Organigramme de l'opérateur de PERWITT.*


Début
Mise à zéro des bords de ImaAux
I = 1
J = 1
Lap1 = 8Ima[I,J]-Ima[I-1,J]-Ima[I+1,J]-Ima[I,J-1]
-Ima[I-1,J-1]-Ima[I+1,J+1]-Ima[I-1,J+1]-Ima[I+1,J-1]
ImaAux[I,J] = [Lap1/16] + [NivMax/2]
J = J+1
J <= MaxCol-1
oui
non
I = I+1
I <= MaxLin-1
oui
non
I = 1
J = 1
Pix = ImaAux[I,J]
PixX = ImaAux[I,J+1]
PixY = ImaAux[I+1,J]
PixXY = ImaAux[I+1,J+1]
((Pix>7) et (PixX<7)) ou ((Pix<7) et (PixX>7)) ou
((Pix>7) et (PixY<7)) ou ((Pix<7) et (PixY>7)) ou
((Pix>7) et (PixXY<7)) ou ((Pix<7) et (PixXY>7)) ou
non
oui
A1


A1
B4
Oui
Max = Pix-PixX
Imal I , J] =0
oui
|Pix-PixY| >Max
non
Max = Pix-PixY
oui
|Pix-PixXY| >Max
Max = Pix-PixXY
Imal I , J] = Max
J = J + 1
J <= MaxCol-1
Oui
B1
I = I + 1
I <= MaxLin-1
Oui
Max = 0
I = 1
B3
J = 1
oui
Imal I , J] > Max
non
Max = Imal I , J]
J = J + 1
J <= MaxCol-1
oui
A2

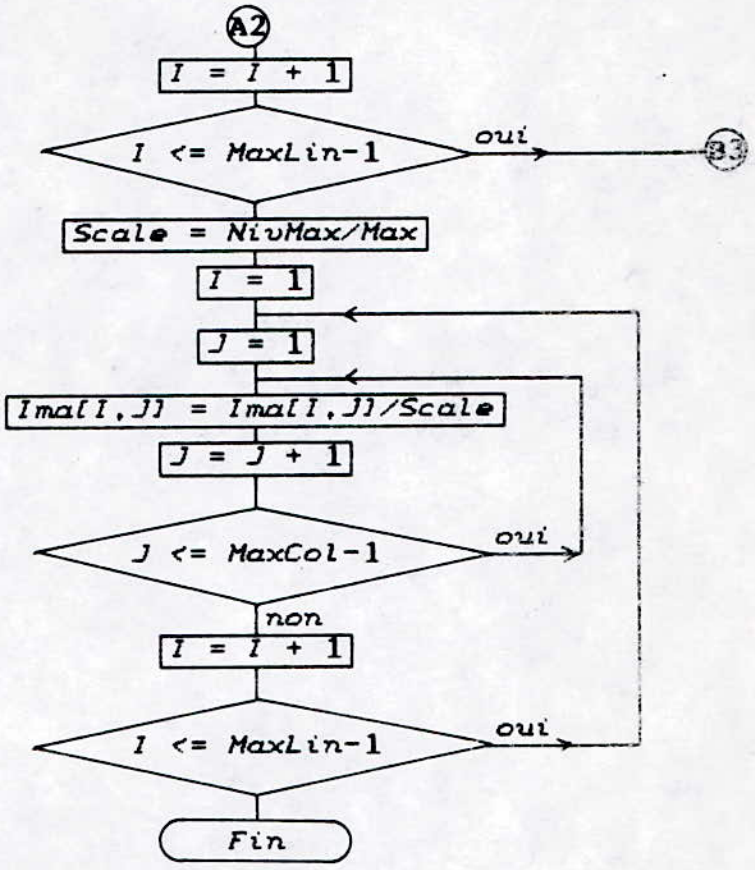
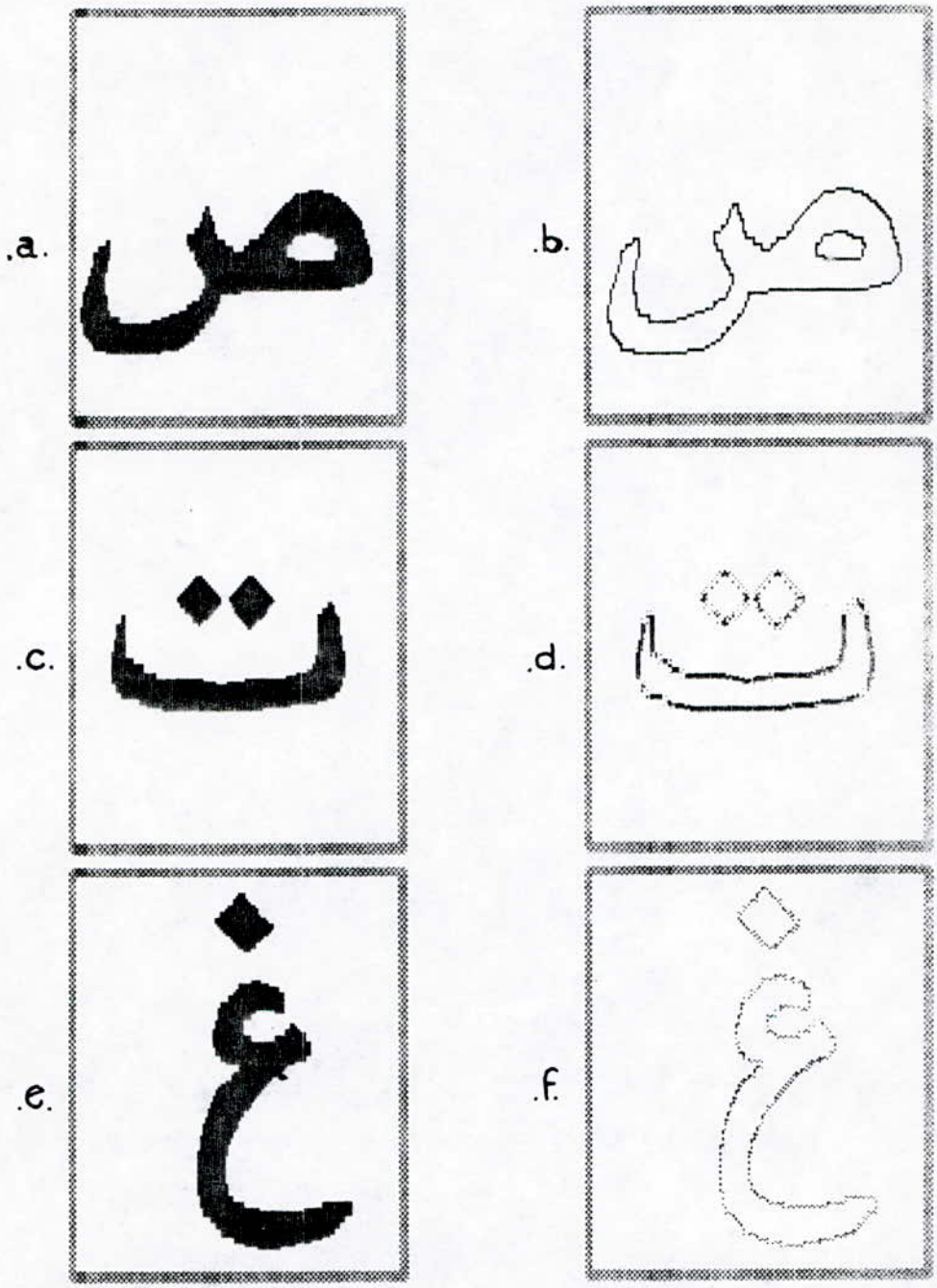
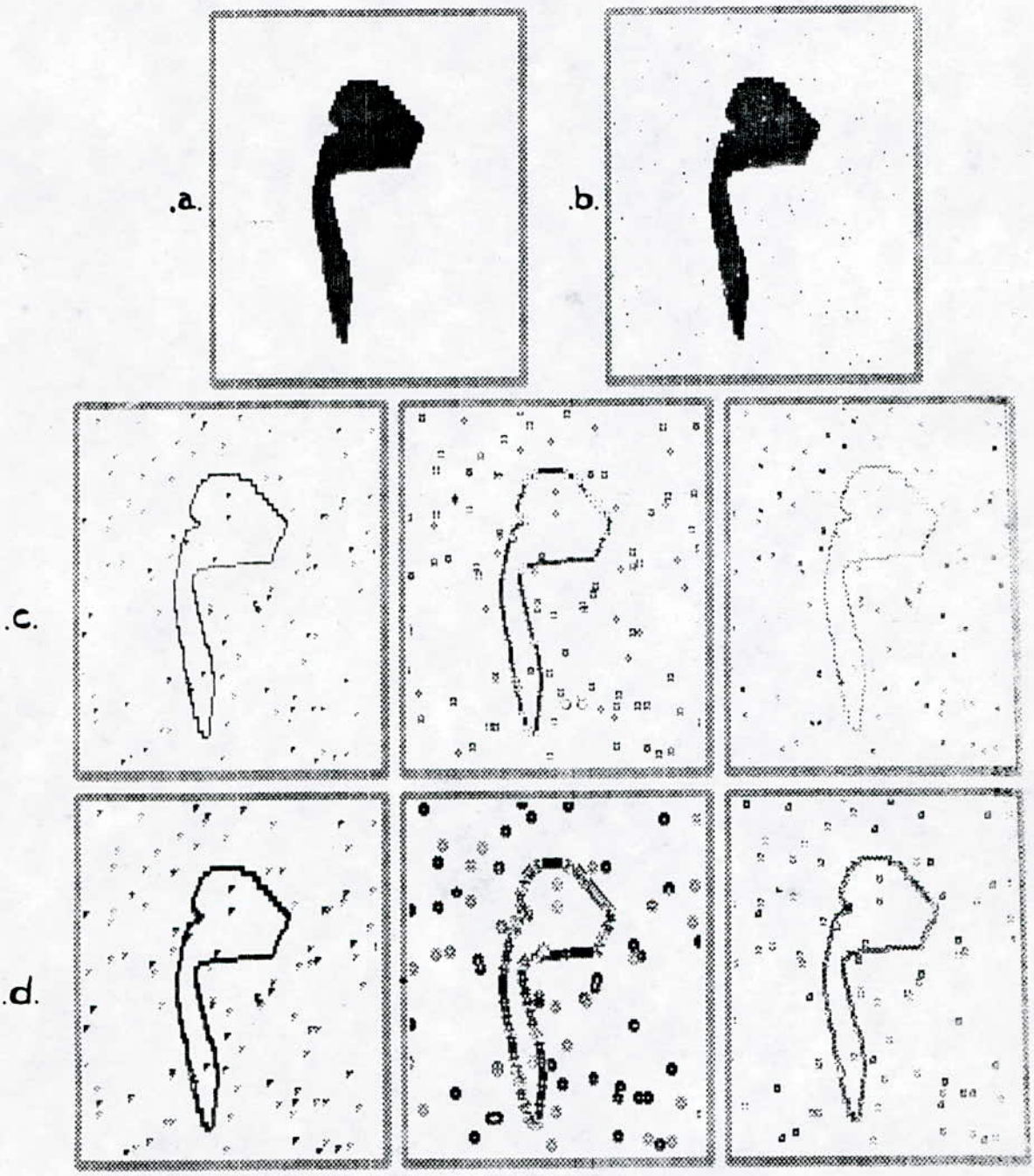


Fig III.6 : *Organigramme de l'operateur de LAPLACE .*

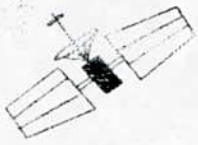
différentiation comme les gradients et le laplacien. Elle a fait l'objet de développements intensifs dans la littérature et chez les industriels du traitement d'image. Les logiciels commercialisés offrent tous, les différents masques que nous avons passé en revue.



Demo III.1. *Detection de contour.(a): Image originale. (b): Contour avec l'operateur de Roberts. (c): Image originale. (d): Contour avec Sobel. (e): Image originale. (f): Contour avec le LAPLACIEN.*



Demo III.2. *Effet du bruit sur la detection du contour. (a): Image originale. (b): Image bruitée (d=145,n=10). (c): $1^{ère}$ derivation avec respectivement les operateurs de Roberts, Sobel et Laplace. (d): $2^{ème}$ derivation avec respectivement Roberts, Sobel et Laplace.*



CHAPITRE IV

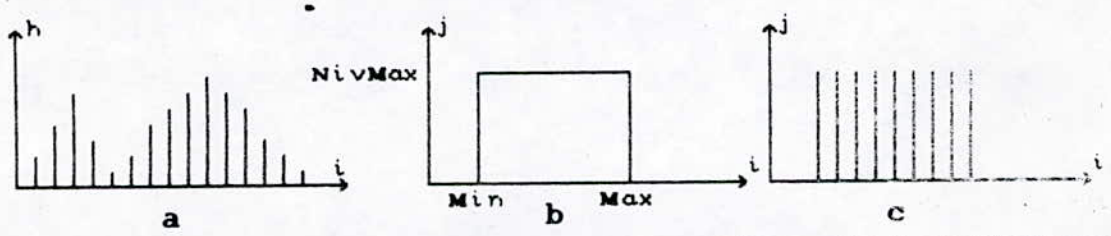
BINARISATION
OPERATEURS MORPHOLOGIQUES

La binarisation d'une image présente plusieurs avantages dont l'un des plus importants est certainement le faible espace mémoire nécessité par une image binaire et la simplicité des opérateurs qui lui sont associés. En effet, l'image possédant que deux niveaux, noir ou blanc, il suffit d'un bit pour coder un pixel d'où une taille mémoire plus réduite. La binarisation peut être aussi, la première étape d'isolement des objets par rapport au fond.

Du fait de leurs codages sur deux niveaux de gris, chaque pixel d'une image binaire peut être codé comme un élément logique, donc à valeur vrai ou faux. Les opérateurs morphologiques sont issus des études sur la morphologie mathématique et reposent sur le concept de transformation géométrique d'une image par un élément structurel.

1.BINARISATION :

Généralement, une binarisation nous permet d'isoler les objets en blanc sur fond noir. Le principe consiste à forcer les pixels, dont le niveau de gris est compris dans l'intervalle [*Min,Max*], à la valeur *NivMax*. Le reste des pixels est forcé à zéro (Voir Fig.IV.1)



FigIV.1.*Binarisation. (a): Histogramme originale. (b): Lut de la binarisation. (c): Histogramme final.*

Il est courant dans un processus de binarisation de n'utiliser qu'un seul seuil (*Min* ou *Max*). Dans ce cas, l'autre est porté à sa valeur extréme (*zero* pour *Min*, *NivMax* pour *Max*). Cette binarisation peut être détérminée automatiquement par recherche du minimum de l'histogramme (Voir Démo.IV.1). Ceci peut être fait en réalisant une interpolation polynômiale de celui ci, puis en dérivant le polynome obtenu afin de déceler le minimum et d'ajouter le seuil en conséquence.

Organigrammme :

L'organigramme de la binarisation est sur la Fig.IV.2.

Variables :

ImaIn.....: Image originale.

ImaOut.....: Image finale.

I,J.....: Indices.

2.OPERATEURS MORPHOLOGIQUES :

L'élément structurant, introduit précédemment, est un masque de forme quelconque dont les éléments forment un motif. L'application de l'opérateur consiste à balayer l'image avec ce masque et à effectuer pour chaque pixel une mise en correspondance du pixel et de ses voisins, avec le motif du masque, puis d'en effectuer *l'union* ou *l'intersection*. Cette technique se rencontre dans la terminologie anglaise sous le nom de *template matching*. Chaque pixel est ainsi l'objet d'une transformation par une fonction logique plus ou moins complexe.(Voir Fig IV.3)

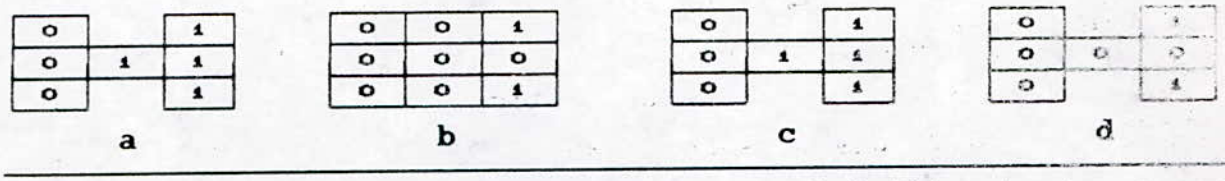


Fig IV.3. *Exemple d'éléments structurant. (a): élément structurant. (b): image originale. (c): union. (d): intersection.*

2.1.DILATATION :

Elle consiste à dilater l'image. De ce fait, les points isolés au milieu des parties blanches sont "mangés" par la dilatation des parties blanches (Voir Démo IV.2).

Le processus est réalisé en balayant l'image avec une fenêtre carrée de taille $(N+1)x(N+1)$ et en effectuant pour chaque pixel de l'image le "ou" logique des $(N+1)^2-1$ voisins c'est à dire tous les pixels de la fenêtre sauf le pixel central. Le résultat de cette fonction est alors: si le ou des voisins vaut 1, alors le pixel courant est forcé à 1 dans l'image résultat ceci est illustré dans la Fig IV.4.

Organigramme :

L'organigramme de la dilatation est illustré sur la Fig IV.5.

Variables :

Ima.....: Image originale.

ImaBin1.....: Image binaire initiale.

ImaBin2.....: Image binaire finale.

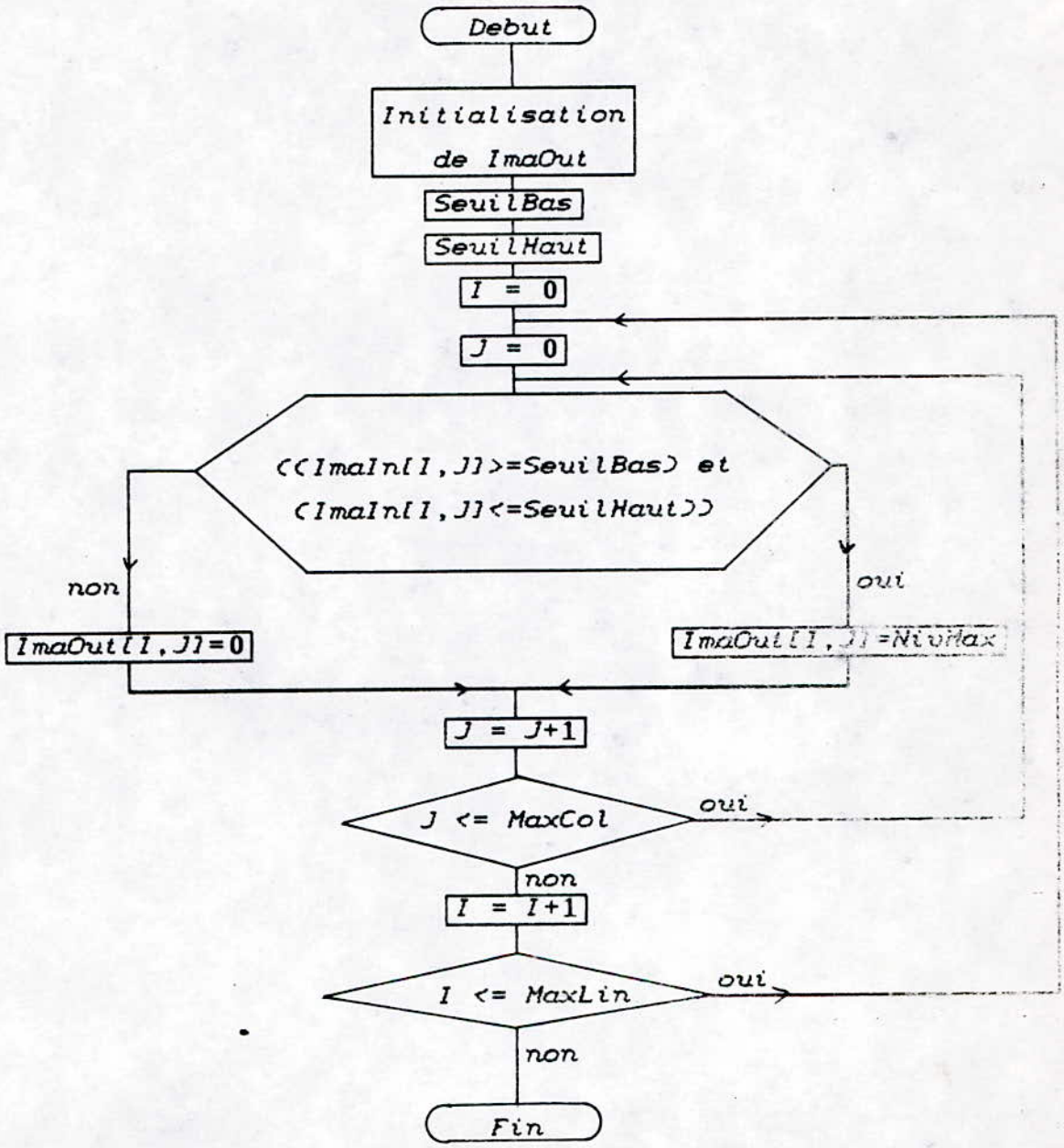


Fig IV.2. *Organigramme de la binarisation*

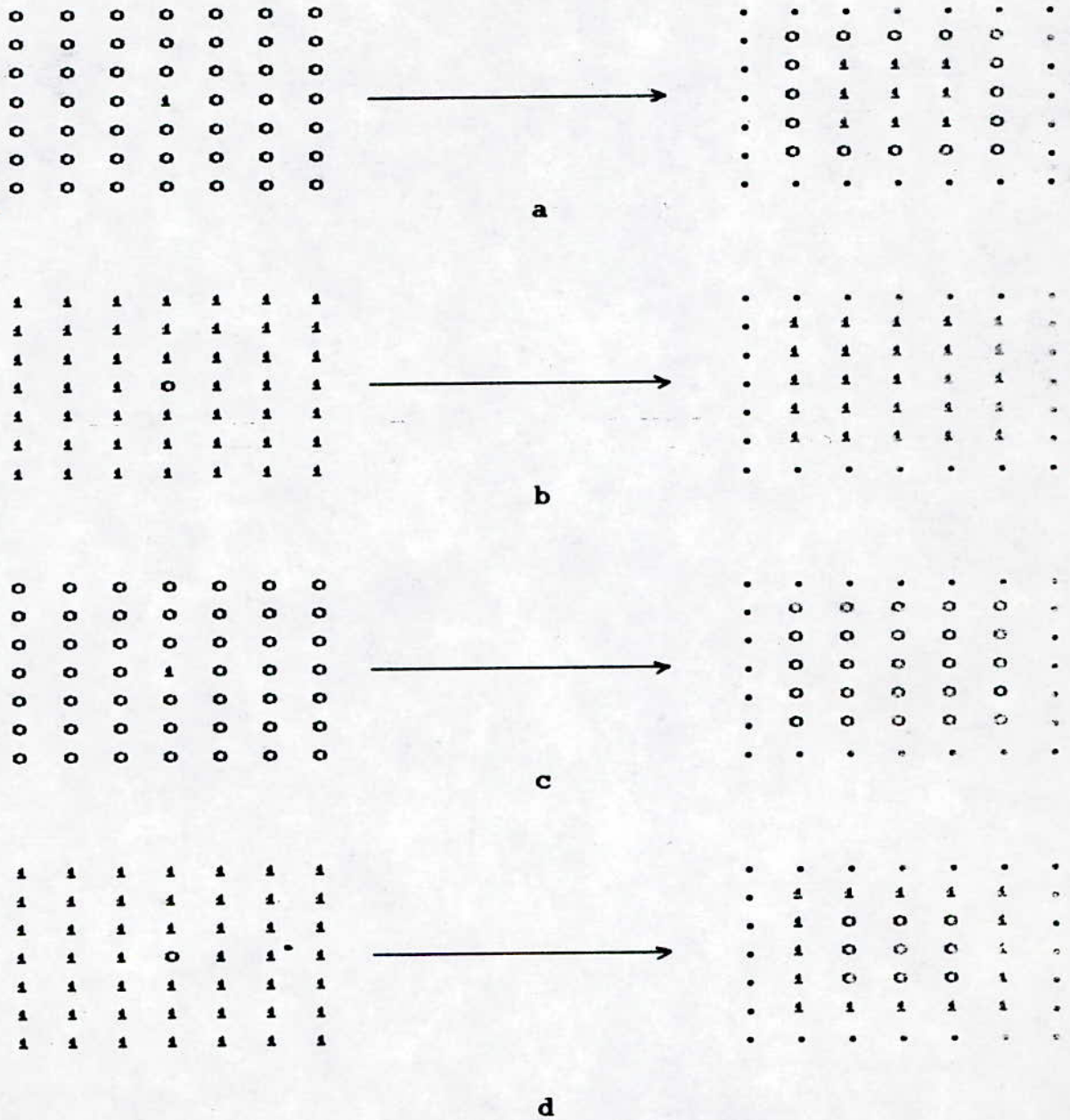


Fig IV.4. *Effets de dilatation et d'érosion. (a, b): Dilatation. (c, d): Erosion.*

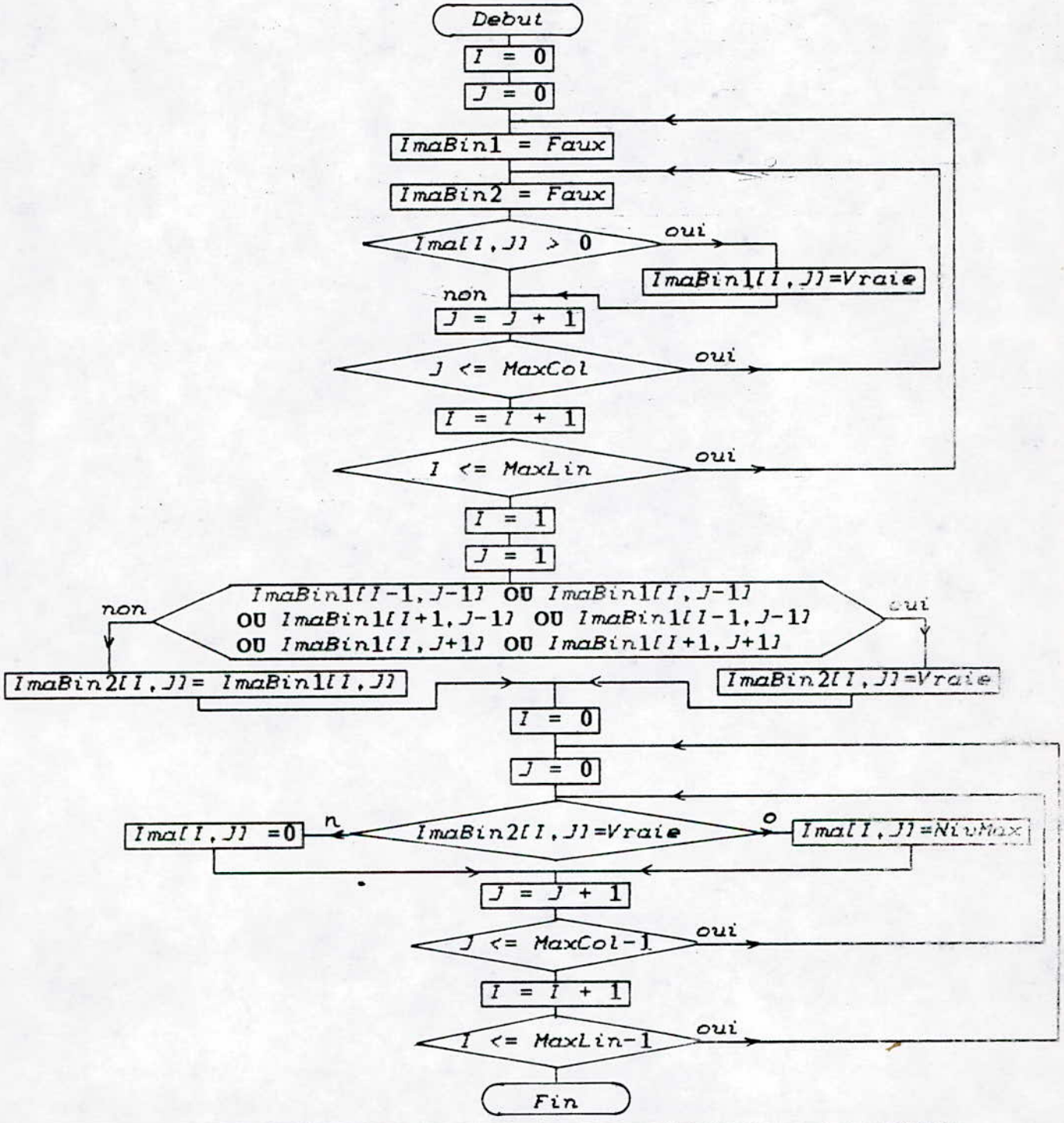


Fig IV.5 *Organigramme de la dilatation.*

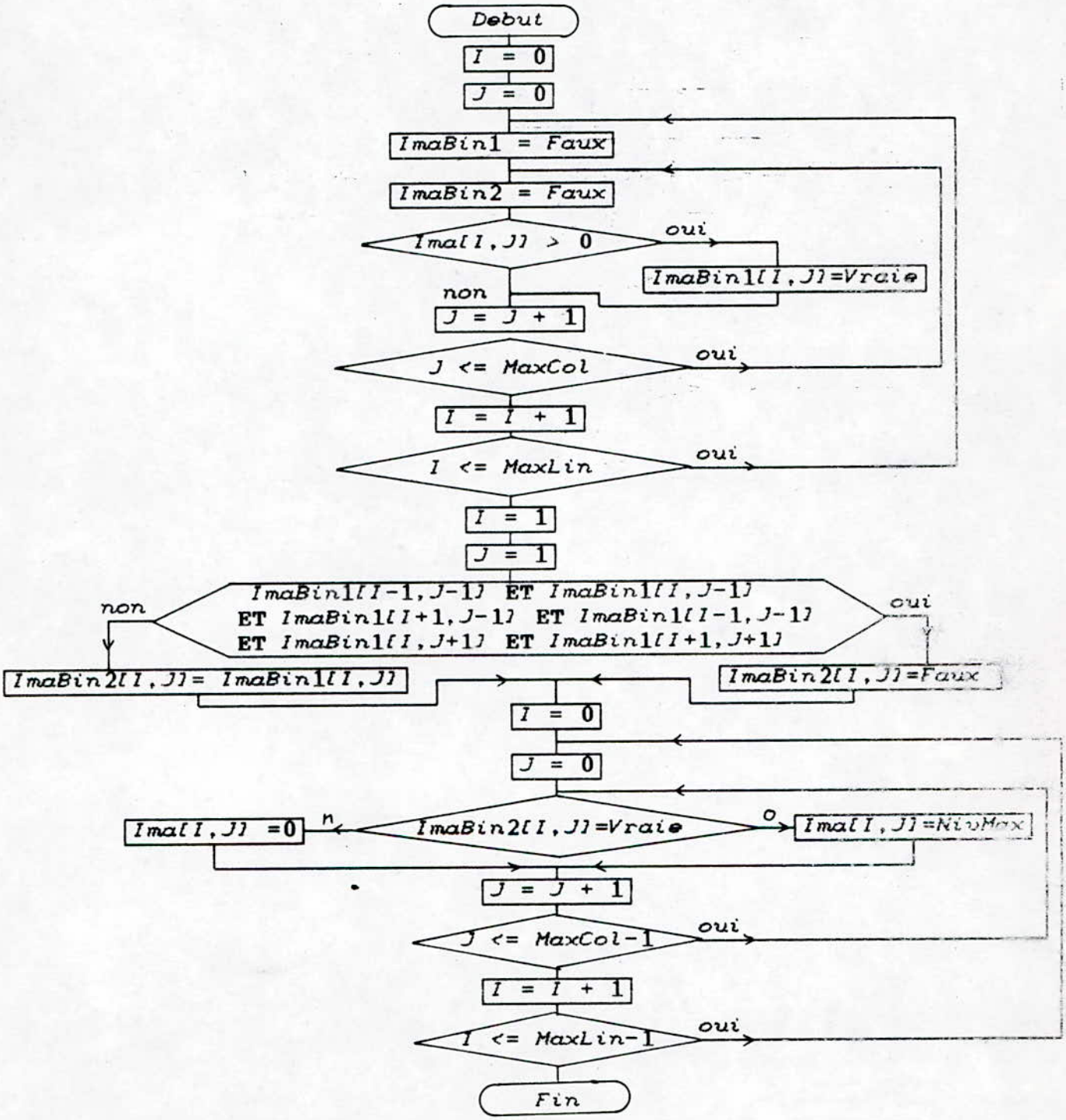


Fig IV.6 *Organigramme de l'erosion.*

2.2. EROSION :

Le procédé est dual à la dilatation. On effectue cette fois le "ET" logique des $(N+1)^2-1$ voisins. De ce fait l'image se trouve érodée, et tout pixel blanc isolé, disparait(voir Démo IV.2). Le principe est illustré sur la Fig IV.4.

Organigramme :

Voir Fig IV.6 .

Variables :

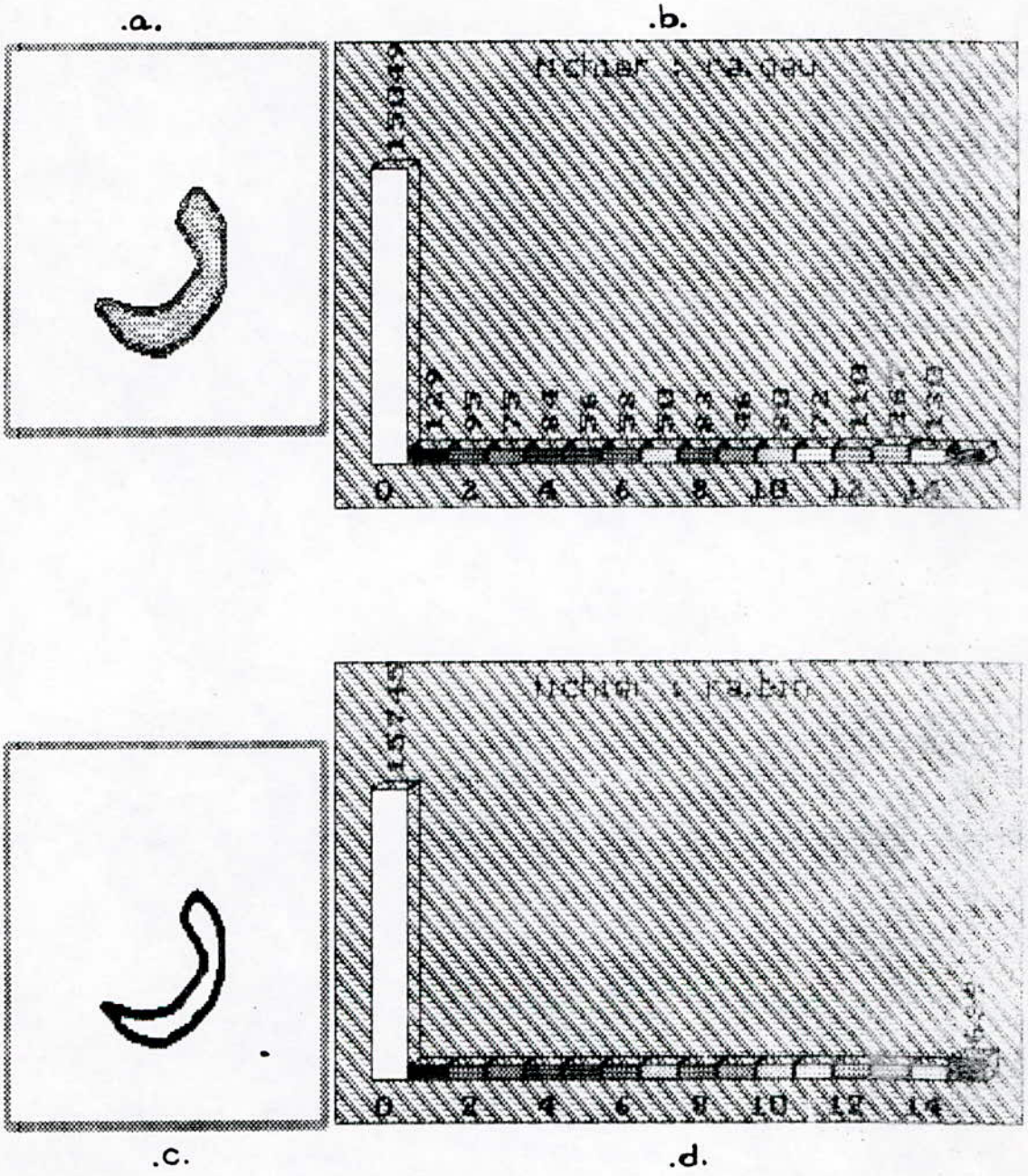
Ima.....: Image originale.

ImaBin1.....: Image binaire initiale.

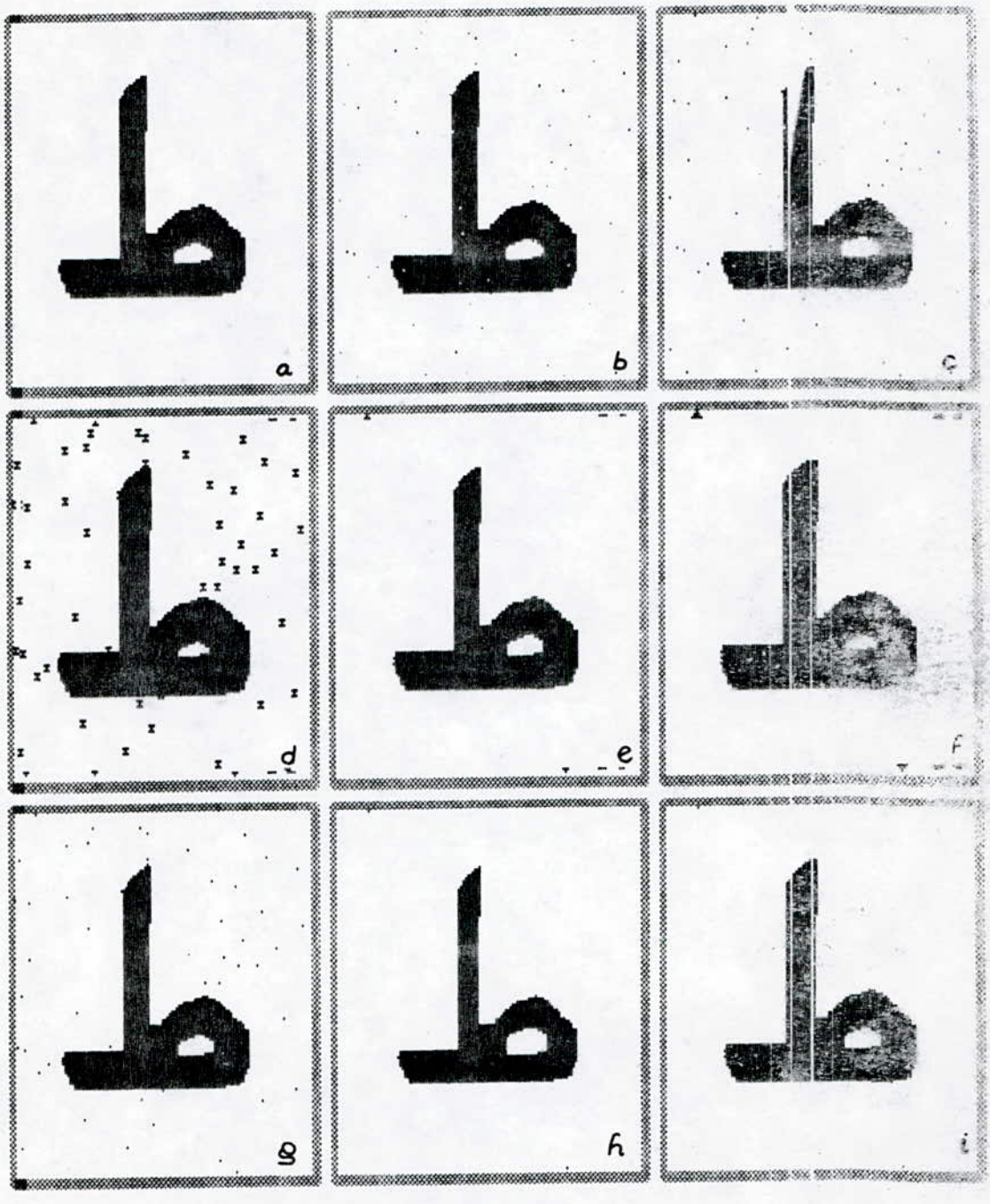
ImaBin2.....: Image binaire finale.

3. CONCLUSION :

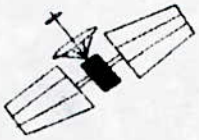
Les tâches noires de l'image peuvent être éliminées par plusieurs dilatations, de même les tâches blanches par plusieurs érosions. Cependant, ces procédures influent sur la taille des objets présents dans l'image. Il est donc nécessaire de faire suivre un nombre D de dilatations par un même nombre d'érosions et réciproquement. En général on effectue une dilatation/érosion suivie d'une érosion/dilatation afin de supprimer les points isolés de l'image. On dépasse rarement ce nombre du fait de la puissance de ces opérateurs qui suppriment une grande quantité d'information dans l'image par la perte des détails de petite taille.



Demo IV.1. *Binarisation. (a,b): Image et son histogramme simple. (c,d) : Image binarisée (Seuil bas=2, seuil haut=10) et son histogramme simple.*



Demo IV.2. *Opérateurs morphologiques. (a): Image originale. (b): Image bruitée (d=220,n=10). (c): Image binarisée entre 1 et 15. (d,e,f): 1ère, 2ème et 3ème dilatation. (g,h,i): 1ère, 2ème et 3ème erosion.*



CHAPITRE V

RECONNAISSANCE

Une fois la segmentation de l'image effectuée, c'est à dire extraire d'une image les formes, qui à ce stade sont definies par leurs primitives de contour. En effet après la segmentation les caractères sont caractèrisées par une sequence de pixels qui appartiennent à leurs contours fermés. pour pouvoir reconnaitre ces caractères, il est nécessaire de placer sur chaque forme une *étiquette*.

Dans ce chapitre on expose un moyen de *décrire* ou modéliser nos images dans le plan afin de leur donner une identité; les caractères ne seront plus des formes géometriques mais deviendrons un *vecteur d'attribute*. Ces derniers sont dans notre cas des descripteurs de contour représentés par une serie de sept(7) moments. Avant, la reconnaissance proprement dite, il est au préalable nécessaire d'en extraire les caractèristiques de tous les caractères : ce qui constitue la phase d'*apprentissage*. La dernière opération importante restant à faire pour arriver à identifier le caractère est la *reconnaissance*.

1. EXTRACTION DES CARACTERISTIQUES :

L'extraction des caractèristiques est le dernier processus de réduction de l'information et est fortement liée à la structure de la bibliothèque des objets que l'on désire pouvoir reconnaitre. En effet, ces caractèristiques doivent avoir la propriété d'effectuer une bonne discrimination d'un objet à l'autre tout en restant en nombre limité afin de réduire les temps de calcul tant au niveau de leur extraction qu'au niveau de leur analyse par le système de reconnaissance. Ceci explique pourquoi les systèmes actuels de vision sont capables de reconnaitre un nombre limité d'objets pour une

application donnée tout en gardant des performances honorables.

On peut classer les caractèristiques suivant trois familles :

* Les caractèristiques topologiques.
* Les caractèristiques fonctionnelles.
* Les caractèristiques géométriques.

Dans notre cas on a à faire à des caractères arabes isolés. Ces derniers sont décrits par une serie de moments, qui font partie des caractèristiques fonctionnelles. On obtient ainsi par décomposition de l'image dans une base de fonctions (moments) une représentation des caractères sous forme de serie. La dimension de la base est sept du fait qu'on s'est limité aux sept premiers termes du développement.

Les Moments :

soit $I(i,j)$ l'image traitée que l'on veut reconnaitre. Le moment généralisé d'ordre $(p+q)$ d'une telle image s'ecrit :

$$m_{pq} = \sum_{i=0}^{Maxlin} \sum_{j=0}^{Maxcol} i^p \, j^q \, I(i,j) \qquad \text{[ref 5]}$$

pour p,q = 0,1,2,3.

Dans une image blanche sur fond noir, le moment d'ordre 0 représente la surface du caractère : $m_{00} = \sum_i \sum_j I(i,j)$

Les moments d'ordre 1 sont définis par :

$$m_{10} = \sum_i \sum_j i \; I(i,j) \qquad m_{01} = \sum_i \sum_j j \; I(i,j)$$

ils définissent le centre de gravité $(\bar{X},\bar{Y})$ du caractère considéré :

$$\bar{X} = \frac{m_{10}}{m_{00}} \qquad \bar{Y} = \frac{m_{01}}{m_{00}}$$

Les *moments centrés* d'ordre (p,q) sont donnés par :

$$\mu_{pq} = \sum_{i=0}^{Maxlin} \sum_{j=0}^{Maxcol} (i-\bar{X})^p (j-\bar{Y})^q \; I(i,j) \qquad \text{[ref 5]}$$

Les moments centrés allant jusqu'a l'ordre trois (3) sont :

$$\mu_{10} = \sum_{i=0}^{Maxlin} \sum_{j=0}^{Maxcol} (i-\bar{X})^1 (j-\bar{Y})^0 \; I(i,j) = m_{10} - \frac{m_{10}}{m_{00}} m_{00} = 0$$

$$\mu_{01} = \sum_{i=0}^{Maxlin} \sum_{j=0}^{Maxcol} (i-\bar{X})^0 (j-\bar{Y})^1 \; I(i,j) = m_{01} - \frac{m_{01}}{m_{00}} m_{00} = 0$$

$$\mu_{11} = \sum_{i=0}^{Maxlin} \sum_{j=0}^{Maxcol} (i-\bar{X})^1 (j-\bar{Y})^1 \; I(i,j) = m_{11} - \frac{m_{10}\, m_{01}}{m_{00}}$$

$$\mu_{20} = \sum_{i=0}^{Maxlin} \sum_{j=0}^{Maxcol} (i-\bar{X})^2 (j-\bar{Y})^0 \; I(i,j) = m_{20} - \frac{m_{10}^2}{m_{00}}$$

$$\mu_{02} = \sum_{i=0}^{Maxlin} \sum_{j=0}^{Maxcol} (i-\bar{X})^0 (j-\bar{Y})^2 \; I(i,j) = m_{02} - \frac{m_{01}^2}{m_{00}}$$

$$\mu_{30} = \sum_{i=0}^{Maxlin} \sum_{j=0}^{Maxcol} (i-\bar{X})^3 (j-\bar{Y})^0 \; I(i,j) = m_{30} - 3\bar{X}m_{20} + 2m_{10}\bar{X}^2$$

$$\mu_{12} = \sum_{i=0}^{Maxlin} \sum_{j=0}^{Maxcol} (i-\bar{X})^1 (j-\bar{Y})^2 \; I(i,j) = m_{12} - 2\bar{Y}m_{11} - \bar{X}m_{02} + 2\bar{Y}^2 m_{10}$$

$$\mu_{21} = \sum_{i=0}^{Maxlin} \sum_{j=0}^{Maxcol} (i-\bar{X})^2 (j-\bar{Y})^1 \; I(i,j) = m_{21} - 2\bar{X}m_{11} - \bar{Y}m_{20} + 2\bar{X}^2 m_{01}$$

$$\mu_{03} = \sum_{i=0}^{Maxlin} \sum_{j=0}^{Maxcol} (i-\bar{X})^3 (j-\bar{Y})^0 \; I(i,j) = m_{03} - 3\bar{Y}m_{02} + 2m_{01}\bar{Y}^2$$

En resumé ;

$$\mu_{oo} = m_{oo} \; ; \qquad \mu_{11} = m_{11} - \bar{Y} m_{10}$$

$$\mu_{10} = 0 \; ; \qquad \mu_{30} = m_{30} - 3\bar{X} m_{20} + 2 m_{10} \bar{X}^2$$

$$\mu_{01} = 0 \; ; \qquad \mu_{21} = m_{21} - 2\bar{X} m_{11} - \bar{Y} m_{20} + 2\bar{X}^2 m_{01}$$

$$\mu_{20} = m_{20} - \bar{X}\, m_{10} \; ; \qquad \mu_{12} = m_{12} - 2\bar{Y} m_{11} - \bar{X} m_{02} + 2\bar{Y}^2 m_{10}$$

$$\mu_{02} = m_{02} - \bar{Y}\, m_{01} \; ; \qquad \mu_{03} = m_{03} - 3\bar{Y} m_{02} + 2 m_{01} \bar{Y}^2$$

Ces moments sont invariants par translation de l'image.

Les *moments centrés normés* notés η_{pq} se defissent par l'expression :

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{oo}^{\gamma}} \qquad \text{avec} \quad \begin{cases} p+q = 2, 3, \ldots \\ \gamma = \dfrac{p+q}{2} \end{cases} \qquad \text{[ref 5]}$$

Ces moments engendrent des moments généralisés invariants par translation, rotation et homothetie, c'est à dire par le groupe des similitudes affines (Voir Démo.V.1). Nous en présenterons ci-dessous un groupe de sept :

$$\phi_1 = \eta_{20} + \eta_{02} \ ;$$

$$\phi_2 = (\eta_{20} + \eta_{02})^2 + 4\eta_{11}^2 \ ;$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} + \eta_{03})^2 \ ;$$

$$\phi_4 = (\eta_{30} - \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \ ;$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \ ;$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \ ;$$

$$\phi_7 = (3\eta_{12} - \eta_{30})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \ ;$$

Il est intéressant de noter que des silhouettes d'avions en vol ont pu être décrites par la mesure des sept moments géneralisés.[ref 3]

II. APPRENTISSAGE :

Comme on l'a dit, avant d'être opérationnel, le processus de reconnaissance doit passer par une phase d'apprentissage. L'apprentissage consiste à établir une correspondance entre le caractère et son modéle. Notre système va en extraire les moments de chaque caractère (de l'alphabet arabe isolé) de manière à constituer une bibliothèque de moments. Cette notion de bibliothèque montre bien que l'on évolue dans un univers limité de caractères reconnaissables. Ceci signifie que si l'on demande par la suite au système de reconnaitre un

caractère qui n'appartient pas à la bibliothèque (ou non isolé) il en sera incapable ou bien donnera une réponse erronnée.

La base de donnée que nous avons élaborée est constituée de 29 caractères isolés. Elle a été integrée dans un fichier portant le nom **"BASE.DAT"** et possède une structure matricielle 29 lignes x 7 colonnes. Chaque ligne présente les caractèristiques (moments) d'un des caractères constituant la base de donnée (Voir Fig V.1).

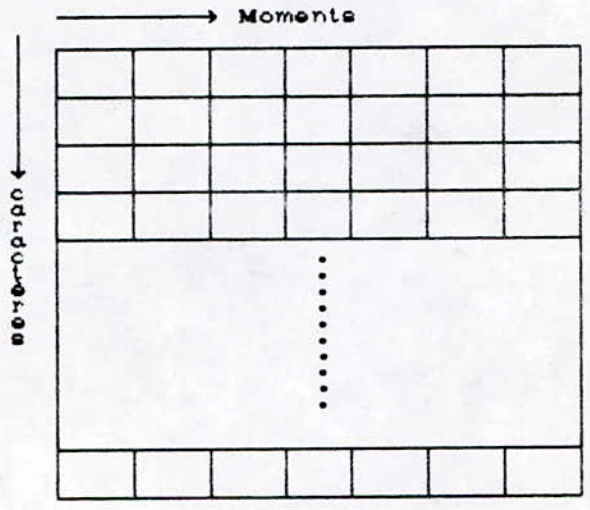


Fig V.1. : *Organisation de la base de données.*

3. RECONNAISSANCE ET DECISION :

L'étape finale de la vision constitue l'interprétation

des caractèristiques extraites de l'image; c'est à dire la reconnaissance des formes.

Le choix d'un processus de reconnaissance de formes est lié à la base au type de déscription que l'on désire pour caractèriser un objet. Il existe grossièrement deux:

- Description paramètrique
- Description structurelle

Le modèle choisi dans notre cas est un modèle paramètrique qui consiste à caractèriser un caractère par un vecteur dont les éléments sont les sept moments invariants décrits précédamment. Le choix d'un tel modèle est dû à sa simplicité de construction ce qui lui a permis une large application. La seule difficulté consiste à choisir une liste de caratèristiques qui permettent une bonne discrimination entre les différents objets de la bibliothèque (Voir Fig V.2 représentant la distribution des moments dans la base de données).

Regle de decision :

Nous avons vu que chaque caractère de la base de données est représentéé par un vecteur S_{ij} i= 1..29, j=1..7, où i designe l'indicé du caractère et j l'indice relatif aux caractèristiques (Ordre du moment).

La régle de décision utilisée s'appuie sur la notion de distance euclidienne. On definit cette distance entre deux points repérés par leurs vecteurs r_1 et r_2 de dimension N par :

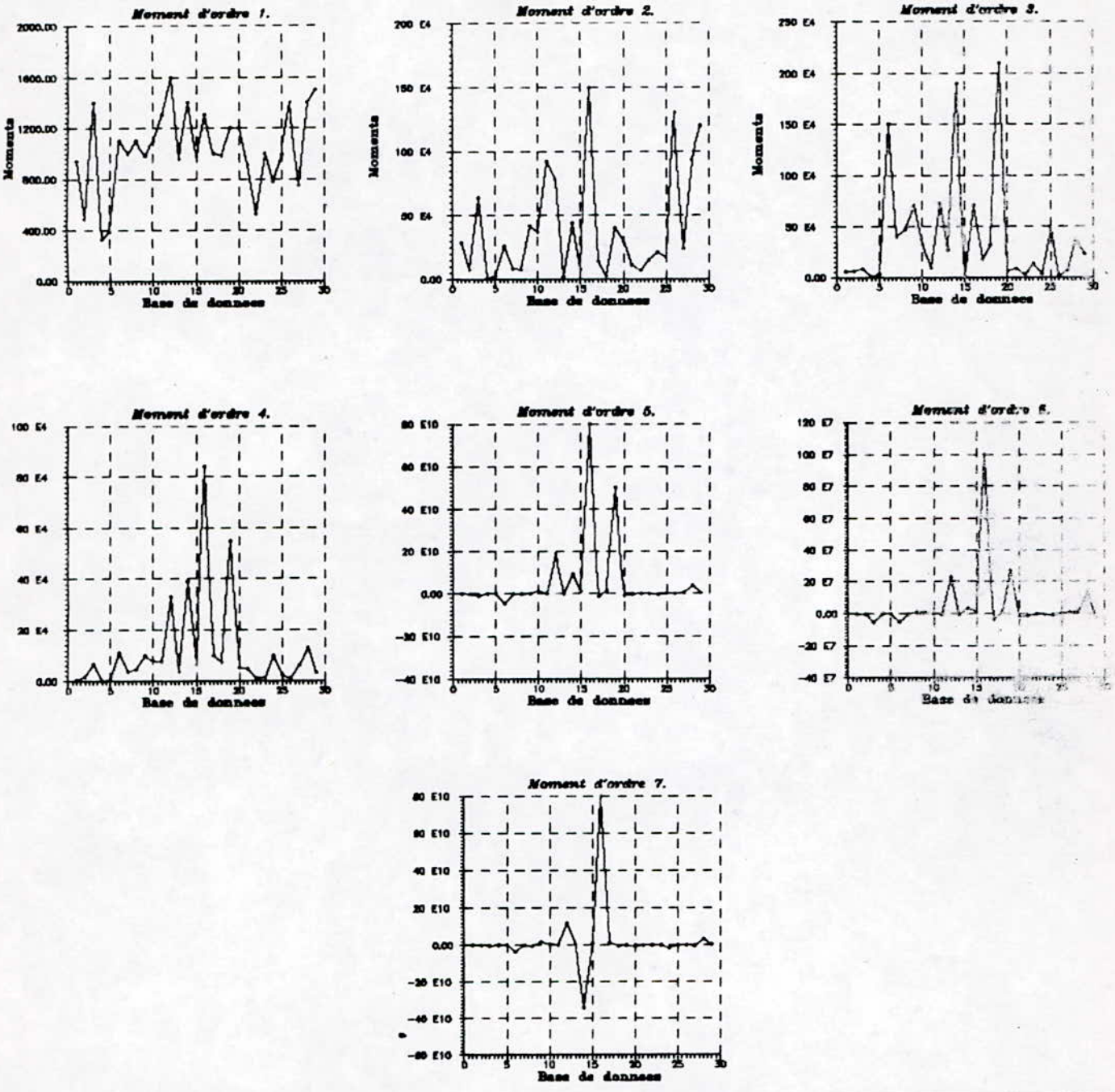


Fig V.2 *Courbes représentant les variations des sept moments. L'axe des X représente les 29 caractéres arabes donnés dans l'ordre suivant :*

$$D(r_1, r_2) = \| r_1 - r_2 \|$$

$$= ((r_1 - r_2)^T (r_1 - r_2))^{1/2}$$

$$= \left[\sum_{k=1}^{N} (r_{1k} - r_{2k})^2 \right]^{1/2}$$

La reconnaissance d'un caractère représenté par un vecteur X consiste à trouver la distance miniale D (S_i, X) définie comme suit :

$$D(S_i, X) = \text{Min} \left\{ D(S_l, X) \right\}_{l=1..20}$$

Donc la i^{eme} ligne de la base de données représente les moments de l'élément le plus voisins du caractère X que l'on veut reconnaitre.

4. RESULTATS OBTENUS :

4.1. Temps moyens :

4.1.1. Temps moyen de filtrage :

Les valeurs données ont été obtenues par une moyenne sur des caractères pris aléatoirement de l'alphabet arabe.

Taille / Filtre	3	5	7	9
Moyen	11.40 "	25.85 "	45.90 "	1'5.40"
Gauss	18.90 "	27.80 "	35.80 "	44.00 "
Median		17.70 s		

4.1.2. Temps moyen de detection de contour :

Opérateurs	*Roberts*	*Sobel*	*Perwitt*	*Laplace*
Resultats	3.15 "	6.60 "	7.10 "	11.40 "

4.1.3. Temps moyen de binarisation :

Opérations	*Binarisation*	*Dilatation*	*Erosion*
Résultats	1.05 "	5.40 "	3.50 "

4.1.4. Temps moyen de calcul des moments :

Des essais sur un échantillon de caractères ont donné le temps moyen suivant :

40,50 "

4.1.5. Temps moyen de calcul du plus proche voisin :

Pour divers caractères, on a obtenu le temps suivant :

5.5 "

On note que ce temps est trés instable, car il dépend de la disposition des moments dans la base de données.

4.1.6. Temps moyen global de reconnaissance automatique :

49.5 "

On precise que la fonction Cettime de Turbo-Pascal utilisée, donne une variation de ∓5 centieme de second.

4.2. Taux de reconnaissance :

4.2.1. Taux de reconnaissance pour caractères rotés :

Angle (rad)	π/2	-π/2	π
Taux (%)	98	98	99

4.2.2. Taux de reconnaissance pour caractères translatés :

Translation (X , Y)	10,10	-10,10	10,-10	-10,-10
Taux (%)	99	99	98	99

Les valeurs de la translation ne sont pas limitées pourvu que le caractére ne sort pas du cadre.

4.2.3. Taux de reconnaissance pour caractères ayant subit une homothétie :

Rapport d'homothetie	0.9	0.97	1.03	1.1
Taux (%)	1	99	98	1

On voit bien que le caractére n'est pas reconnu lorsque le rapport d'homothécie est hors de l'intervalle [0.97,1.03]. A cause des erreurs d'arrondissements.

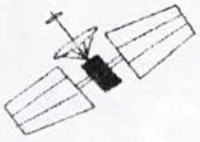
4.2.4. Taux de reconnaissace pour caractères ayant subi une symetrie :

Axe de symetrie	X	Y	XY
Taux (%)	99	99	98

4.2.5. Taux globale de reconnaissance :

95 %

Demo V.1. *Invariance des moments pour differentes rotations et symetries*



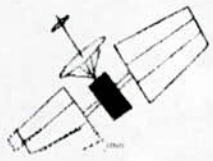
CONCLUSION

Dans la presente étude, nous avons developpé un logiciel qui nous a permis d'étudier la methode des moments pour la reconnaissanse des caractères arabes isolés. Elle ,nous a permis de voir les principales techniques du traitement d'image. Les caractères utilisés sont des caractères eus par scaner (TIF) ce qui a permis de donner un aspect plus ou moins réaliste à notre étude.

L'objectif visé est largement atteint : il s'agissait d'étudier les performances des moments sur les caractères arabes isolés. Les sept moments invariants ont été suffisants pour modeliser les caractères avec une discrimination acceptable. En effet, le taux de reconnaissance globale, qui est de 95 % , est acceptable avec un temps de calcul relativement bon (50"). La methode s'est avérée efficace pour les caractères ayant subi des translations, rotations et symetries, mais moins performante pour l'homothétie à cause des erreurs d'arrondie qui interviennent dans le calcul de cette transformation. Ce qui n'est pas le cas pour une homothètie réelle (non simulée).

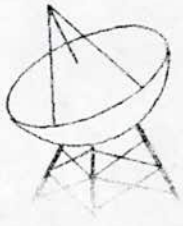
En ce qui concerne le filtrage, l'operateur est invité à choisir le filtre le plus adapté à son application. Par contre, pour la détection de contour, le choix s'est porté sur l'opérateur de Laplace qui est le plus approprié, dans notre cas, pour la reconnaissance.

Le produit réalisé, présente l'avantage d'être réalisé sur matériel compatible, d'où la possibilité d'une large utilisation. On espére qu'il fournira une main d'aide dans le domaine de la *vision assistée par ordinateur* V . A . O .



ANNEXE I

PRINCIPALES PROCEDURES PASCAL REALISEES



PRINCIPALES DECLARATIONS ET PROCEDURES UTILISEES

```
unit  Declarations;
interface
uses crt,graph,dos;
const
   maxlin=127;
   maxcol=127;
   nivmax=15;
   nbrelet=29;
type
   Mom=array[1..7] of real;
   Base=array[1..nbrelet] of Mom;
   FileMom=File of Mom;
   FileBase=File of Base;
   lin=array[0..maxlin] of byte;
   image=array[0..maxcol] of lin;
   fimage=file of image;
   nomfich=string[20];
   linlon=array[0..maxcol] of integer;
   imalon=array[0..maxlin] of linlon;
   flinlon=file of linlon;
   fimalon=file of imalon;
   imabin=array[0..maxlin,0..maxcol] of Boolean;
var
    ImaOut,ImaIn            :image;
procedure SaveIma(nom:nomfich;var ima:image);(*sauvegarde une image*)
procedure ChargeIma(nom: nomfich;var ima:image;var tem:boolean);
procedure Affima(imag:image;xie,yje:integer);
procedure DetruitDmage(ch:nomfich);
procedure CreeImage(var ch:nomfich);
procedure lecgr(var strg:string);(* lit un string *)
procedure lecnbr(var nbr:integer;message:string);(* lit un nombre *)
procedure TraceHisto(var imag:image;X,Y:integer;sg:nomfich);
procedure TraceHistoCum(var imag:image;X,Y:integer;sg:nomfich);
procedure Efface_Fenetre;
procedure ROBERTS(var CH:NOMFICH);
procedure Sobel(var ch:nomfich);
procedure Perwitt(var ch:nomfich);
procedure LAPLACE(var ch:nomfich);
procedure Binair(ch:nomfich);
procedure Dilatation(Ch:NomFich);
procedure Erosion(ch:nomfich);
procedure Moyenne(var ch:nomfich);
procedure EgalHis(var nf:nomfich);
procedure Gauss(var ch:nomfich);
procedure Median(var ch:nomfich);
procedure IntroBruit(var ch:nomFich);
procedure Renomer;
procedure lgr(mi:string;var strg:string); (* lit un string *)
procedure Impr; (* imprime l'écran de sortie *)
procedure Auto (var ch:nomfich); (* reconnaissance automatique *)
begin
end.
```

Cette procedure calcul le nombre de niveaux de gris reellement present dans l'image, puis effectue l'égalisation en divisant ce nombre par deux pour obtenir le nombre de niveaux fineaux.

```
procedure moyenne(var ch:nomfich);
label 22;
var
       taille,demi_taille   :integer;
       ImaI,ImaO            :image;
       i,j,k,l              :integer;
       scale,Somme          :integer;
       boo                  :boolean;

 begin
   chargeIma(ch,ImaI,boo);
   if boo=false then goto 22;
   settextstyle(2,0,5);setcolor(7);
   lecnbr(taille,'Taille du filtre en pixel : "Taille>3 ;Taille<127");
   scale:=taille*taille;
   demi_taille:=(taille-1) div 2;

   {i n i t i a l i s a t i o n   t a b l e a u}

   for i:=0 to maxlin do
     for j:=0 to maxcol do
       ImaO[i,j]:=0;

   efface(49);outtextxy(50,getmaxy-22,' Calcul en cours.... ');

  (*   F  I  L  T  R  A  G  E  *)

  for i:= demi_taille  to  Maxlin-demi_taille do
  for j:= Demi_taille  to  Maxcol-demi_taille do

   begin

         somme:=0;
         for k:=i-demi_taille  to i+demi_taille  do
          for l:=j-demi_taille  to j+demi_taille  do

            begin
              somme:= somme + imaI[k,l];textbackground(14);
            end;

          imao[i,j]:= somme div scale;
   end;

   repeat until keypressed;

22: end;
```

Cette procedure réalise un filtrage passe bas. La taille du masque doit etre impaire et peut etre comprise entre 0 et 127.

```
procedure egalhis(var nf:nomfich);
label 22;
const   nivmax=15;
type
     hist=array[0..255] of integer;
var    imin,imout              :image;
       histo,transf            :hist;
       gd,gm,pixel,moyen       :integer;
       bande,centrebande,ec    :integer;
       somme,sommeprec         :integer;
       OrigNiv,Finniv          :integer;
       i,j,ideb,iprec          :integer;
       boo                     :boolean;

 begin
       chargeima(nf,ImIn,boo);if boo=false then goto 22;
       gettime(h0,m0,t0,hd0); t0:=3600*h0+60*m0+t0;
       redfich(nf);
       for i:=0 to nivmax do
        histo[i]:=0;

(********************** Calcul   Histograme  **********************)

settextstyle(2,0,5);setcolor(7);
efface(49);outtextxy(50,getmaxy-22,' Calcul en cours ... ');
     for i:=0 to maxlin do
      for j:=0 to maxcol do
        begin
          pixel:=imin[i,j];
          histo[pixel]:=histo[pixel]+1;
        end;
(*** Calcul du nombre de niveaux de gris reelement present dans l'image
 OrigNiv:=0;
 for i:= 0 to nivmax do
   if histo[i]<>0 then  origniv:=origniv+1;

 (**CALCUL DE NOMBRE DE NIVEAUX FINAUX ET LARGUEUR DE CHAQUE BANDE FINALE

   finniv:=(origniv div 1);
   bande:=nivmax div finniv;
   centrebande:=(bande div 2);
   moyen:=(((maxlin+1)*(maxcol+1)) div finniv);

 (***** CALCUL DE LA TABLE DE TRANSFORMATION *****)

    i:= 0 ; ideb:=0 ; iprec :=0;
    while i <= nivmax do

   (*** INTEGRATION DES NIVEAU TANT QUE LA VAL MOY PAS DEPASSE *****)

 begin
  SOMME:=0;
```

```
     WHILE (SOMME< MOYEN) and (i<= nivmax) do
      begin
           sommeprec:= somme;
           somme:=somme+histo[i];
           if (somme<moyen) and (histo[i]>0) then    iprec:=i;
           i:=i+1;
      end;

(***ON REGARDE QUELLE SOMME EST LA PLUS PROCHE DE LA VALEUR IDEALE***)

      if abs(somme-moyen)<=abs(sommeprec-moyen)  then
    begin
      for j:=ideb to i-1 do  transf[j]:=centrebande;
      ideb:=i;
    end
      else
    begin
       for j:= ideb to iprec do begin transf[j]:=centrebande;end;
       ideb:=iprec+1;
    end;

  (****************** ON PASSA A LA BANDE SUIVANTE ***********************

       centrebande:=centrebande + bande;
       iprec:=i;
       if centrebande> nivmax then begin
       writeln;gotoxy(20,20);
       efface(49);outtextxy(50,getmaxy-22,'niveau 15 depassé...:');
       end;
   end;

{ ///////////////*** Egalisation ***\\\\\\\\\\\\\\\\\\\}

      for i:=0 to maxlin do
      for j:= 0 to maxcol do
        begin
             pixel:=ImIn[i,j];ImOut[i,j]:=transf[pixel] ;
        end;
       repeat until keypressed;
22:  end;
```

Cette procedure assure un filtrage gaussien d'une image en utilis-ant la separabilité du filtre. La taille du filtre est parametrable et doit etre impaire, comprise entre 3 et 127.

```
procedure gauss(var ch:nomfich);
label 22;
var
   filtre                        :array[0..128] of real;
   taillefiltre,demitaille       :integer;
   sigma,somme                   :real;
```

```
    ima,imatemp                    :image;
    s0                             :string[10];
    i,j,k,l                        :integer;
    pix1,pix2                      :real;
    boo                            :boolean;

begin
  chargeima(ch,ima,boo);if boo=false then goto 22;
     settextstyle(2,0,5);setcolor(7);
     lecnbr(taillefiltre,'Taille du filtre en pixel (impaire)  : ');
     gettime(h0,m0,t0,hd0); t0:=3600*h0+60*m0+t0;
     sigma:=(taillefiltre-1) / 4;writeln;
     efface(49);outtextxy(50,getmaxy-22,' sigma = '{+str(sigma,s0)});
     demitaille:=(taillefiltre-1) div 2;

      (*..... coeficient......*)
     efface(49);outtextxy(50,getmaxy-22,' Calcul en cours ... ');
     somme:=0;
     for i:=0 to taillefiltre-1 do
      begin
          filtre[i]:=exp(-((i-demitaille)*(i-demitaille)/2./sigma/sigma));
          somme:=somme+filtre[i]
      end;

      for i:=0 to taillefiltre-1 do
        filtre[i]:=filtre[i]/somme;

(***************INITIALISATION DES DIFFERENTS TABLEAUX*********************

      for i:=0 to maxlin do
        for j:=0 to maxcol do
          imatemp[i,j]:=0;
                                    101
(**********************FILTRAGE SUIVANT LES COLONNES**************************

     for i:=demitaille to maxlin-demitaille do
       for j:=demitaille to maxcol-demitaille do
          begin
             pix1:=0;
               for k:= i-demitaille to i+demitaille do
                 begin
                    pix2:=ima[k,j];
                    pix1:=pix1+(filtre[k-i+demitaille]*pix2)
                 end;
             imatemp[i,j]:=trunc(pix1);
          end;

(**********************MISE A ZERO DE L'IMAGE RESULTAT************************

         for i:=0 to maxlin do
           for j:=0 to maxcol do
             ima[i,j]:=0;

(**********************FILTRAGE SUIVANT LES LIGNES ***************************
```

```
      for i:=demitaille to maxlin-demitaille do
       for j:=demitaille to maxcol-demitaille do
        begin
          pix1:=0;
           for k:=j-demitaille to j+demitaille do
             begin
              pix2:=imatemp[i,k];
              pix1:=pix1+(filtre[k-j+demitaille]*pix2)
             end;
          ima[i,j]:=trunc(pix1);
        end;
repeat until keypressed;
 22:  end;
```

Cette procedure réalise un filtrage median d'une image avec un masque en croix de taille 5. L'algorithme de tri utilisé est de type methode de la BULLE.

```
procedure median(var ch:nomfich);
label 22;
var imain,imaout              :image;
    i,j,k,l,inter             :integer;
    tri                       :array[0..8] of integer;
    endtri                    :boolean;
    boo                       :boolean;

begin
     setcolor(7);
     chargeima(ch,imain,boo);if boo=false then goto 22;
     settextstyle(2,0,5);setcolor(7);
     efface(49);outtextxy(50,getmaxy-22,' Calcul en cours ... ');

(*************** MISE A ZERO DU CADRE ****************)

for i:=0 to maxlin do
 for j:=0 to 1 do
   begin
    imaout[i,j]:=0;
    imaout[i,maxcol-j]:=0;
   end;
 for i:=0 to 1 do
   for j:=0 to maxcol do
     begin
      imaout[i,j]:=0;
      imaout[maxlin-i,j]:=0;
     end;
 outtextxy(50,getmaxy-22,' Calcul en cours ... ');
 for i:=2 to maxlin-2 do
  for j:=2 to maxcol-2 do
   begin

(****REMPLISSAGE DE LA TABLE DE TRI A L'AIDE DES PIXELS VOISINS DU PIXEL**
(*************COURANT COMPRIS DANS LA CROIX DE LA TAILLE 5*5**************
```

```
    for k:=-2 to 2 do tri[k+2]:=imain[i,j+k];
    for k:=-2 to -1 do tri[k+7]:=imain[i+k,j];
    for k:=1  to 2  do tri[k+6]:=imain[i+k,j];

(*************TRI DESPIXELS DE LA TABLE PAR ORDRE CROISSANT**************

    endtri:=false;
    while not(endtri) do
      begin
       endtri:=true;
       for l:=1 to 8 do
        if tri[l-1]>tri[l] then
          begin
           inter:=tri[l];
           tri[l]:=tri[l-1];
           tri[l-1]:=inter;
           endtri:=false;
          end
      end;

    (*****************CHOIX DES VALEURS MEDIAN ************************)

      imaout[i,j]:=tri[4]
   end;
  repeat until keypressed;
22: end;
```

Cette procedure genere un bruit impulsionnel et l'ajoute à l'image de travail. Ce bruit est paramètrable en intensité et en densité.

```
PROCEDURE IntroBruit(var ch:nomFich);
label 22;
var
   ImaIn,ImaOut              :image;
   i,j,c1,c2                 :integer;
   pixel,pixBruit,bruit      :integer;
   Densite                   :integer;
   NivBruit1,Nivbruit2       :integer;
   boo:boolean;

begin
    ChargeIma(ch,ImaIn,boo);if boo=false then goto 22;
    randomize;

    (*INTRODUCTION DS PARAMETRES*)

    setcolor(7); settextstyle(2,0,5);
    outtextxy(50,getmaxy-22,'INTRODUCTION DE BRUIT DANS L''IMAGES' );
    delay(1000);efface(0);
    lecnbr(NivBruit1,'Niveau du bruit  : ');
    lecnbr(Densite,'Densite de bruit : ');

    (* introduction du bruit *)
```

```
    efface(49);outtextxy(50,getmaxy-22,' Calcul en cours...  ');
    for i:=0 to maxlin do
      for j:=0 to maxcol do
       begin

     (* DETERMINATION DE L'AFFECTATION DE BRUIT SUR UN PIXEL OU NON *)

         Pixel:=random(Densite)+1;

         (* PIXEL AFFECTE PAR LE BRUIT *)

         if Pixel=Densite then
          begin
             Bruit:=random(NivBruit1);
             PixBruit:=ImaIn[i,j]+Bruit;
             if PixBruit>NIVMAX then PixBruit:=0;
             if PixBruit<0 then PixBruit:=nivmax;
             ImaOut[i,j]:=PixBruit;
          end

         (* PIXEL NON AFFECTE PAR LE BRUIT *)

         else
              ImaOut[i,j]:=ImaIn[i,j]
       end;
      repeat until keypressed;
22:end;
```

Cette procedure calcul le gradient d'une image en utilisant l'opér -ateur de Roberts. Le résultat est un tableau contenant les amplitudes du gradient. Celle-ci sont calculées par le max des valeurs en X et Y.

```
PROCEDURE ROBERTS(var CH:NOMFICH);
label 11;
var
     imain,imaout     :image;
     GradX,GradY      :integer;
     i,j              :integer;

begin
       chargeima(ch,imain,bop);if bop=false then goto 11;
       settextstyle(2,0,5);        efface(49);
       outtextxy(50,getmaxy-22,' Calcul en cours... ');
       for i:=0 to maxlin do
        for j:=0 to 1       do
         begin
          imaout[i,j]:=0;
          imaout[i,maxcol-j]:=0;
         end;
      for i:=0 to 1 do
       for j:=0 to  maxcol do
        begin
           imaOut[i,j]:=0;
           imaout[maxlin-i,j]:=0;
        end;

(*          D  E  R  I  V  A  T  I  O  N          *)
   for i:= 1 to maxlin-1 do
     for j:=1 to maxcol-1 do
      begin
        gradX:=ImaIn[i,j]-ImaIn[i,j-1];
        gradY:=imain[i,j]-imain[i-1,j];
{*** Calcul  de l''amplitude par le max des valeurs absolues  ***}
        if abs(gradx)<abs(grady)  then imaout[i,j]:=abs(grady)
        else imaout[i,j]:=abs(gradx);
      end;
repeat until keypressed;
11:end;
```

Cette procedure calcul le gradient d'une image en utilisant l'opér -ateur de Sobel. Le résultat est un tableau contenant les amplitudes du gradient. Celle-ci sont calculées par le max des valeurs en X et Y.

```
procedure sobel(var ch:nomfich);
label 11;
var
     Imain,ImaOut   :image;
     Gradx,GradY    :integer;
     i,j            :Integer;
 begin
    chargeima(ch,ImaIn,bop);if bop=false then goto 11;
```

```
    settextstyle(2,0,5);
    efface(49);outtextxy(50,getmaxy-22,' Calcul en cours... ');

 {   mise   à   zero   du   cadre...}

  for i:=0 to maxlin do
   for j:=0 to 1 do
     begin
          ImaOut[i,j]:=0;
          Imaout[i,maxcol-j]:=0;
     end;
   for i:=0 to 1 do
    for j:=0 to maxcol do
      begin
           ImaOut[i,j]:=0;
          ImaOut[maxlin-i,j]:=0;
      End;
    for i:=1 to maxlin-1 do
     for j:=1 to maxcol-1 do
      begin
       GradX:= ImaIn[i-1,j+1]+2*ImaIn[i,j+1]+imain[i+1,j+1]
              -Imain[i-1,j-1]-2*Imain[i,j-1]-imain[i+1,j-1];
       Grady:= ImaIn[i+1,j-1]+2*ImaIn[i+1,j]+imain[i+1,j+1]
              -Imain[i-1,j-1]-2*Imain[i-1,j]-imain[i-1,j+1];
       if abs(GradX)<abs(gradY) then ImaOut[i,j]:=abs(GradY) div 4
                                else ImaOut[i,j]:=abs(gradX) div 4;
      end;
   repeat until keypressed;
11:  end;
```

Cette procedure calcule le Laplacien d'une image. L'image resultat est une matrice des passages par zéro du laplacien. Les valeurs sont recadrées entre zéro et NivMax.

```
PROCEDURE LAPLACE(var ch:nomfich);
label 11;
var
    ima,ImaAux        :image;
    i,j,lap1,max,pix :integer;
    pixx,pixY,pixXY   :integer;
    scale             :real;

Begin
      chargeima(ch,Ima,bop);if bop=false then goto 11;
      settextstyle(2,0,5);

(*  MISE A ZERO DES BORDS    *)

efface(49);outtextxy(50,getmaxy-22,' Calcul en cours...');
   for i:=0 to maxlin do
     for j:=0 to 1 do
      begin
          imaAux[i,j]:=0;
          imaAux[i,maxcol-j]:=0;
```

```
      end;
    for i:=0 to 1 do
      for j:=0 to maxcol do
       begin
           ImaAux[i,j]:=0;
           ImaAux[maxlin-i,j]:=0;
       end;

{************ DERIVATION resultat ds ImaAux, 0<val ************}
{********       negative<126, 128<val positive<255 *************}
for i:=1 to maxlin-1 do
 for j:=1 to maxcol-1 do
  begin
   lap1:=8*ima[i,j]-ima[i-1,j]-ima[i+1,j]-ima[i,j-1]-ima[i,j+1]
                   -ima[i-1,j-1]-ima[i+1,j+1]-ima[i-1,j+1]-ima[i+1,j-1];
    imaAux[i,j]:=lap1  + 120
  end;

{ D E T E C T I O N  D E S  P A S S A G E S  P A R  Z E R O  }
for i:=1 to maxlin-1 do
 for j:=1 to maxcol-1 do
   begin
     pix:=imaAux[i,j];
     pixX:=imaAux[i,j+1];
     pixy:=imaAux[i+1,j];
     pixXY:=imaAux[i+1,j+1];
     if (((pix>120) and (pixX<120)) or ((pix<120) and (pixX>120)) or
       ((pix>120) and (pixY<120)) or ((pix<120) and (pixY>120)) or
       ((pix>120) and (pixXY<120)) or ((pix<120) and (pixXY>120)))
    then
      begin
       Max:=abs(pix-pixX);ima[i,j]:=max;
  if abs(pix-pixY) > Max then begin
                                Max:=abs(pix-pixY);ima[i,j]:=max;
                              end;
  if abs(pix-pixXY) > Max then begin
                                Max:=abs(pix-pixXY);ima[i,j]:=max;
                               end;
      end
  else ima[i,j]:=0;
end;

{ calcul de l'amplitude max pour recadrer le resultat entre 0 et 255 }
max:=0;
for i:=1 to maxlin-1 do
 for j:=1 to maxcol-1 do
  if ima[i,j]> max then max:=ima[i,j];

(******************** R E C A D R A G E  ****************)
  scale:= nivmax/max;
for i:=1 to maxlin-1 do
 for j:=1 to maxcol-1 do
 ima[i,j]:=trunc(ima[i,j]*scale);
repeat until keypressed;
11: end;
```

Cette procedure réalise la binarisation de l'image. L'utilisateur est invité à entrer les seuils bas et hauts de binarisation.

```
procedure Binairisation(ch:nomfich);
label 22;
var
  ImaIn,ImaOut          :image;
  i,j                   :integer;
  seuilBas,SeuilHaut    :integer;
  boo                   :boolean;
begin
for i:=0 to maxlin do
  for j:=0 to maxcol do
     Imaout[i,j]:=0;
chargeIma(ch,ImaIn,boo);if boo=false then goto 22;
  begin

(*      le choix des seuils de binarisation      *)
   lecnbr(SeuilBas,' Seuil Bas : ');
   lecnbr(SeuilHaut,' Seuil Haut: ');
   settextstyle(2,0,5);setcolor(7);
   efface(49);outtextxy(10,getmaxy-22,'Calcul en cours . . .');

   (***********binarisation de l'image binaire*************)

      for i:=0 to maxlin do
        for j:=0 to maxcol do
         if (Imain[i,j]>=SeuilBas) and (ImaOut[i,j]<=seuilHaut)
         then
          begin
           ImaOut[i,j]:=nivmax;
          end
        else
         begin
           ImaOut[i,j]:=0;
         end;
   repeat until keypressed;
 22:end;
end;
```

Cette procédure effectue une dilatation de l'image en utilisant un masque de taille fixe 3x3.

```
procedure Dilatation(ch:NomFich);
label 22;
VAR
   ima                 :image;
   Imabin1,imaBin2     :imaBin;
   i,j                 :integer;
   boo                 :boolean;
begin
  chargeIma(ch,Ima,boo);if boo=false then goto 22;
  settextstyle(2,0,5);setcolor(7);efface(49);
  outtextxy(50,getmaxy-20,' Calcul en cours... ');
```

```
(*********  INITIALISATION DES DIFFERENTS TABLEAUX **********************
(*********IMAGE INITIALE ImaBin1 RESULTAT DANS ImaBin2*****************

 for i:=1 to maxlin-1 do
  for j:=1 to maxcol-1 do
   begin
     ImaBin1[i,j]:=false;
     ImaBin2[i,j]:=false;
     if Ima[i,j]>0 then ImaBin1[i,j]:=true
   end;

  (******************DILATATION********************)

  for i:=1 to maxlin-1 do
    for j:=1 to maxcol-1 do
     if      ImaBin1[i-1,j-1]     or ImaBin1[i,j-1]
           or  ImaBin1[i+1,j-1] or ImaBin1[i-1,j+1]
           or  ImaBin1[i,j+1]   or  ImaBin1[i+1,j+1]
     then ImaBin2[i,j]:=true
     else ImaBin2[i,j]:=ImaBin1[i,j];

(*   TRANSFERT DANS Ima POUR SAUVEGARDE *)

    for i:=1 to maxlin-1 do
      for j:=1 to maxcol-1 do
        if ImaBin2[i,j] then
         begin
          Ima[i,j]:=nivmax;
         end
        else
        begin
          Ima[i,j]:=0;
        end;

         repeat until keypressed;
      22:end;
```

Cette procédure effectue une érosion de l'image en utilisant un masque de taille fixe 3x3.

```
procedure Erosion(ch:nomfich);
label 22;
var
  Ima             :image;
  ImaBin1,ImaBin2:imabin;
  i,j             :integer;
  boo             :boolean;
begin
       chargeIma(ch,Ima,boo);if boo=false then goto 22;
       settextstyle(2,0,5);setcolor(7); efface(49);
       outtextxy(50,getmaxy-25,' Calcul en cours ... ');

    (****INITIALISATION DES DIFFERENTS TABLEAUX IMAGE INITIALE DANS ****
    (******ImaBin1    RESULTAT DANS ImaBin2 ***************************)
```

```
    for i:=0 to maxlin do
      for j:=0 to maxcol do
       begin
        ImaBin1[i,j]:=false;
        ImaBin2[i,j]:=false;
        if Ima[i,j]>0 then ImaBin1[i,j]:=true
       end;

      (**********EROSION*******************)

      for i:=1 to maxlin-1 do
        for j:=1 to maxcol-1 do

        if     ImaBin1[i-1,j-1]
         and ImaBin1[i,j-1]
         and ImaBin1[i+1,j-1]
         and ImaBin1[i-1,j+1]
         and ImaBin1[i,j+1]
         and ImaBin1[i+1,j+1] then ImaBin2[i,j]:=ImaBin1[i,j]

      else ImaBin2[i,j]:=false;

      (* TRANSFERT DANS Ima POUR SAUVEGARDE*)

      for i:=1 to maxlin-1 do
        for j:=1 to maxcol-1 do
          if ImaBin2[i,j] then
           begin
            Ima[i,j]:=nivmax;
           end
          else
           begin
            Ima[i,j]:=0;
           end;

           repeat until keypressed;
22:end;
```

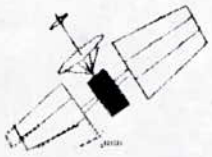
Cette procedure permet de calculer le plus proche voisin du caractère à reconnaitre.

```
procedure distance(harf:NomFich);
label 4,100,11,12;
var tab             :Base;
    dist            :array[1..nbrelet]of real;
    vect            :mom;
    d1,d2           :real;
    i,j,k,h         :integer;
    ha              :NomFich;
    dist_min        :real;
    dst,pb          :string;
    bom             :boolean;
begin   h:=1;
{********** lecture du nom du fichier moment à reconnaitre **********}
11: lgr('le nom du fichier à reconnaitre : ',harf);
{***** lecture du fichier moment et mise des valeurs dans VECT ******}
    lect_mom(harf,vect,test);if not(test) then
         begin
          efface(0);
          outtextxy(10,getmaxy-22,'Autre fichiers O/N :');
          4: case  readkey of
          'o','O':goto 11;
          'n','N':goto 12;
          else goto 4;
         end;end;
{** Lecture de la base de données et mise des valeurs dans TAB ***}
    lect_base('bas.dat',tab,test);
    for i:=1 to nbrelet do  {X=nombre de caracteres}
      begin
        d2:=0;d1:=0;
        for j:=1 to 7 do   {7=nombre de moments}
          begin
            d1:=tab[i,j]-vect[j];
            d2:=d2+sqr(d1);
          end;
      dist[i]:=sqrt(d2);
      end;
      dist_min:=dist[1];
   for i:=1 to 29 do
  begin if dist[i]<dist_min then begin h:=i;dist_min:=dist[i];end;end;
  str(dist_min,dst); str(h,pb);
  outtextxy(10,getmaxy-22,'dist/min = '+dst+'.position dans BD:'+pb);
  outtextxy(10,getmaxy-22,' Afficher la lettre resultat O/N :');
 100:case readkey of
      'o','O': begin dst:='a:\image\'+letre[h]+'.let';
                      chargeima(dst,imaout,bom);
                      case bom of
                       true:affima(imaout,i,j);end;end;
      'n','N':begin end;
      else goto 100;end;
repeat until keypressed;
12:end;
```

Cette procedure permet de calculer les sept moments à partir du contour d'une image.

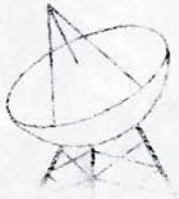
```
procedure moment(var mo:nomfich);
label 1,2,3;
var mi:nomfich;
(***********************************)
function pu(a:real;b:real):real;
 begin
 if a=0 then pu:=0
 else
    pu:= exp( b*ln(a));
 end;
(***********************************)
function pui(a:real):real;
 begin
    pui:=a*a;
 end;
(***********************************)
begin
  chargeima(mo,imain,bo);if bo=false then goto 2;
 efface(49);settextstyle(2,0,5);
outtextxy(20,getmaxy-22,'Fichier :'+mo+'.   Calcul en cours ...');

(***************MOMENTS GENARALISES*********************)

 for p:=0 to 3 do
  for q:=0 to 3 do
     begin
        m[p,q]:=0;
         for i:=0 to maxlin do
     for j:=0 to maxcol do
      begin
        if imain[i,j]<>0 then
        m[p,q]:=m[p,q] + pu(i,p)*pu(j,q);
      end;
     end;
 Xbar:=m[1,0]/m[0,0];
 Ybar:=m[0,1]/m[0,0];

 (**************MOMENTS CENTRES ****************)
  u[0,0]:=m[0,0];
  u[1,0]:=0;
  u[0,1]:=0;
  u[2,0]:=m[2,0] - Xbar*m[1,0];
  u[0,2]:=m[0,2] - Ybar*m[0,1];
  u[1,1]:=m[1,1] - Ybar*m[1,0];
  u[3,0]:=m[3,0] - 3*Xbar*m[2,0] + 2*(Xbar*Xbar)*m[1,0];
  u[1,2]:=m[1,2] - 2*Ybar*m[1,1] - Xbar*m[0,2] + 2*(Ybar*Ybar)*m[1,0];
  u[2,1]:=m[2,1] - 2*Xbar*m[1,1] - Ybar*m[2,0] + 2*(Xbar*Xbar)*m[0,1];
  u[0,3]:=m[0,3] - 3*Ybar*m[0,2] + 2*(Ybar*Ybar)*m[0,1];

 (*************MOMENTS CENTRES NORMALISES*****************)
         for p:=0 to 3 do
          for q:=0 to 3 do
```

```
            begin
              k:=(p+q)/2;
              if ((p+q)>1) then n[p,q]:=u[p,q]/(pu(u[0,0],k));
            end;

 (***************MOMENTS INVARIANTS*********************)
 f[1]:=n[2,0]+n[0,2];
 f[2]:=pui((n[2,0]-n[0,2])) + 4*pui(n[1,1]);
 f[3]:=pui((n[3,0]-3*n[1,2])) + pui((3*n[2,1]+n[0,3]));
 f[4]:=pui((n[3,0]+n[1,2])) + pui((n[2,1]+n[0,3]));
 f[5]:=(n[3,0]-3*n[1,2])*(n[3,0]+n[1,2]) * ( pui((n[3,0]+n[1,2]))-3*
        pui((n[2,1]+n[0,3])) ) + (3*n[2,1]-n[0,3])*(n[2,1]+n[0,3]) *
        (3*pui((n[3,0]+n[1,2]))-pui((n[2,1]+n[0,3])) );
 f[6]:=(n[2,0]-n[0,2])*(pui((n[3,0]+n[1,2]))-pui((n[2,1]+n[0,3])) )
       +4*n[1,1]*(n[3,0]+n[1,2])*(n[2,1]+n[0,3]);
 f[7]:=(3*n[1,2]-n[3,0])*(n[3,0]+n[1,2]) * ( pui((n[3,0]+n[1,2]))-3*
        pui((n[2,1]+n[0,3])) ) + (3*n[2,1]-n[0,3])*(n[2,1]+n[0,3]) *
        (3*pui((n[3,0]+n[1,2]))-pui((n[2,1]+n[0,3])) );
repeat until keypressed;
end;
```



ANNEXE 2

PRISE EN MAIN
DU LOGICIEL REALISE



La version du logiciel donné ,appelé *VISION 1.0*, et qui tourne sur IBM-PC et compatible a été ecrite à l'aide du TURBO PASCAL 5.5/6.0.

Les fichier importants de vision 1.0 :

VISION.EXE : Principal fichier du logiciel.

**.CHR, *.BGI* : Pour la séléction du police d'écriture et du pilote graphique.

BASE.DAT : Constitue la base de données du système.

**.LET* : Fichiers contenant les 29 caractères de réference.

*GRAPHICS.** : Fichiers nécessaires pour l'impression.

La structure d'acceuil :

Sa présentation permet de visualiser trois parties distinctes :

* *Le menu des commandes principales* : Il affiche les commandes principales de **VISION 1.0** .

* *La fenêtre de sortie* : Elle permet de visualiser la sortie à l'écran.

* *La fenêtre des commentaires* : Elle permet de visualiser les commentaires, et d'introduire les données.

1. Le menu des commandes principales :

Le menu principal comporte sept commandes :

ficHier Image Contour Filtre Binarisation Outil Vision

ficHier : Regroupe les opérations sur les fichiers (détruire, renommer) et les répertoires (affichage des fichiers du répertoire, changement du répertoire actif).

Image : Permet d'afficher et de créer une image, ainsi que l'affichage d'histogramme, l'affichage de texte et l'effacement de l'écran.

Contour : Permet de détecter le contour à l'aide des différents opérateurs.

Filtre : Effectue le filtrage de l'image à l'aide du filtre choisi, ainsi que l'introduction de bruit dans l'image.

Binarisation : Permet de réaliser la binarisation et les opérations morphologiques (dilatation, érosion) sur l'image.

Outil : Permet de réaliser les différentes similitude planes sur l'image (translation, rotation, homothétie) et les différentes symétries.

Vision : Regroupe les opérations concernant la reconnaissance (cacul, affichage des moments et la reconnaissance proprement dite), ainsi que l'impression de l'écran.

2. La fenêtre de sortie :

La fenêtre de sortie contient toutes les dernières sorties sur écran générées par les commandes exécutées. Une fois à l'interieur de la fenêtre de sortie, vous pouvez déplacer le témoin en utilisant les touches du curseur gauche, droit, haut et bas. Une fois validée par <Return>, la positon du témoin représente selon le cas le coin haut droit de l'affichage, ou l'un des coins de la fenêtre à effacer.

3. La fenêtre de commentaires :

Dés que vous exécutez une commande, des commentaires apparaissent sur cette fenêtre.

LES COMMANDES DU MENU PRINCIPAL :

Vous pouvez activer une commande du menu de trois manières :

* Utilisez les touches du curseur, gauche et droite, pour vous positionner sur la commande sélectionnée. Validez en pressant la touche <Return>.

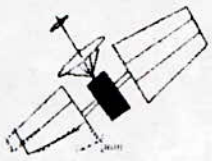
* Tapez la lettre écrite en majuscule à la commande sélectionnée.

* Maintenez la touche <Alt> et tapez la lettre écrite en majuscule à la commande sélectionnée.

L'utilisation du menu déroulant suppose la connaissance des deux remarques suivantes :

1. Pour choisir une option d'un menu déroulant quelconque, vous devez sélectionner avec les touches du curseur haut et bas avant de valider par <Return>.

2. Quelque soit le menu déroulant actif, pressez <Esc> pour revenir au menu immédiatement supérieur ou maintenez la touche <Alt> et tapez la lettre majuscule correspondant à la commande sélectionnée pour passer au menu déroulant de la nouvelle commande choisie.



ANNEXE 3

BIBLIOGRAPHIE

[1] - D.H. Balland and C.M. Brown; *"Computer vision"*; Prentice-Hall 1980.

[2] - J.E. Besançon; *"Vision par ordinateur en deux et trois dimensions"*; Eyrolles 1988.

[3] - S.A. Dudani et al; *"Aircraft identification by moments invariants"*; IEEE Trans. on Comput., Vol C-26 n°1, Jan 1977.

[4] - R.C. Gonzalez et P.Wintz; *"Digital Image Processing"*; Addision-Wesley 1977.

[5] - M.K Hu; *"Visual patten recognition by moments invariants"*; I.R.E Trans. inf. Theory, IT-8, Feb 1962.

[6] - M. Kunt ,*"Traitement numérique des signaux"*; Dunod 1981.

[7] - T. Lachand-Robert ,*"Graphisme en Turbo Pascal"*; Sybex 1989.

[8] - J.J. Toumazet ,*"Traitement de l'image sur micro-ordinateur"*; Sybex 1989.