

الجمهورية الجزائرية الديمقراطية الشعبية  
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

ECOLE NATIONALE POLYTECHNIQUE

Département ELECTRONIQUE

الدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDES

Pour l'Obtention du Diplome d'Ingenieur d'Etat en Electronique

Theme

TRAITEMENT D'IMAGES  
NUMERIQUES.

Dirigé par :

Mr L. SAADAoui

Réalisé par :

Mr T. BENZADI

PROMOTION : Juillet 1994

E.N.P 10, Avenue Hacem Badi EL-HARRACH -ALGER-

MINISTERE DE L'EDUCATION NATIONALE  
ECOLE NATIONALE POLYTECHNIQUE  
DEPARTEMENT ELECTRONIQUE  
PROMOTEUR : Mr L. SAADAoui  
ELEVE INGENIEUR : T. BENZADI

وزارة للتربية و للتعليم  
المدرسة الوطنية المتعددة التقنيات  
دائرة الإلكترونيك  
الأستاذ الموجه : إلياس سعداوي  
صاحب الأطروحة : طارق بن زادي

ملخص :

الهدف من هذا العمل هو دراسة خورزميات علاج الصورة العددية بعدة طرق (تحويل الصورة العددية) ومعرفة دور الخورزميات المطبقة على الصور ذات البنية الثنائية، وترتيب الفعالة منها حسب محافظتها على الإيصال، حفاظها على الهيئة الأصلية للصورة وخلاتها من النقاط الإضافية.

RESUME :

L' Objet de ce travail consiste a faire une etude d' Algorithme de traitement d'images numeriques par plusieurs methodes differentes (transformation d' images binaires), et de voir l'utilite de ces algorithmes d'amincissement et d'epaississement appliques sur des images binaires, et de selectionner ceux qui maintiennent la connexite, preservent la forme originale de l'image et evitent les problemes des pixels supplementaire (cerifs).

SUBSTRACT :

This works aims to investigate the utility of thinning and thickening algorithms to binary pictures, and to select algorithms that will maintain the connectivity, preserve shape of the original image and avoid such problems as extraneous pixels (branches).

## REMERCIEMENTS

Cette thèse est le fruit d'une année d'étude sous la direction de Monsieur L.SAADAOUI.

Je tiens à exprimer ma profonde gratitude à Monsieur SAADAOUI LLIES, qui a su susciter en moi un grand intérêt pour le traitement de l'image en général, et pour ce sujet en particulier, et qui me fait l'honneur de suivre cette étude.

Je remercie également tout particulièrement Madame HAMAMI qui a eu la patience de lire et de corriger une partie de cet ouvrage.

Que tous ceux qui ont participé de près ou de loin à la réalisation de ce projet trouve ici ma sincère gratitude.

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

## PRESENTATION DU SUJET

## PRESENTATION DU SUJET

Le travail consiste essentiellement à étudier le développement des algorithmes d'amincissement et d'épaississement d'images binarisées.

La squelettisation est largement utilisée dans l'analyse des images et la reconnaissance des formes, spécialement pour la réduction de l'espace mémoire réservé au stockage de la structure essentielle de l'information présente dans l'image. Une des méthodes pour l'obtention du "squelette", consiste à effacer des points du bord de l'objet dans l'image sans modifier les caractéristiques topologiques de cette dernière. Plusieurs algorithmes d'amincissement sont présentés et comparés dont deux algorithmes d'amincissement parallèle basés, l'un sur les opérations locales pour détecter les points du bord, les points finaux et les points de rupture, l'autre est l'algorithme de comparaison (matching algorithme) où un ensemble de huit fenêtres et deux images (l'image courante et l'image traitée) sont utilisées dans le processus. de même un algorithme d'épaississement est présenté dans cette étude.

Un algorithme d'amincissement est considéré utile pour des applications particulières si :

- 1- Il préserve la forme de l'image originale.
- 2- Il préserve la connexité.
- 3- Il ne produit pas de cerifs.

Plusieurs études ont été concentré sur deux approches fondamentales. La transformation axe médian (MAT) et la transformation distance (DT). La solidité de la (MAT) est sa capacité de produire des images squelettes toujours connexes. la (DT) par contre peut produire des squelettes ayant des vides pouvant être traités par ordinateur.

L'Objective de ce travail est d'examiner l'utilité des algorithmes d'amincissement et d'épaississement appliqués sur des images binaires, et de sélectionner ceux qui préserve la connexité, la forme de l'image originale et évite le problème de cerifs.

## SOMMAIRE

N° Pages

Dédicaces .....	
Remerciements .....	01
Présentation du sujet .....	02
Introduction générale .....	07
 <b><u>CHAPITRE 1 : ILLUSTRATION - DEFINITIONS et PROCEDURES</u></b>	
1.1- Introduction.....	08
1.2- Illustration générale.....	08
1.3- Définitions.....	11
1.3.1- Les Images Discrètes.....	11
1.3.2- Les Images Binaires.....	11
1.3.3- Le Module d'acquisition.....	12
1.4- Procédures.....	12
1.4.1- Critère de squelettisation.....	12
1.4.2- Les Opérations locales.....	13
1.4.3- Le Processus séquentiel.....	14
1.4.4- Le Processus parallèle.....	14
 <b><u>CHAPITRE 2 : ALGORITHMES D'AMINCISSEMENT APPLIQUES AUX MODELES BINAIREs</u></b>	
2.1- Introduction.....	16
2.2- Définitions.....	16
2.2.1- La Connexité.....	17
2.2.2- Le Point final et le point isole.....	18
2.2.3- Le Point de rupture.....	19

**CHAPITRE 3 : ETABLISSEMENT DES ALGORITHMES DE COMPARAISON ET  
DES OPERATIONS LOCALES**

3.1- Algorithme de comparaison.....	20
3.1.1- Introduction.....	20
3.1.2- L'algorithme d'ARCELLI.....	20
3.1.3- Résultats.....	22
3.2- Algorithme d'amincissement parallèle.....	23
3.2.1- Introduction.....	23
3.2.2- L'algorithme de NACCACHE.....	23
3.2.3- Analyse de l'algorithme.....	26
3.2.4- Résultats.....	28
3.3- La version simplifiée de HILDITCH.....	28
3.3.1- Description.....	28
3.3.2- Analyse de l'algorithme.....	29
3.3.3- Résultats.....	29

**CHAPITRE 4 : ALGORITHME D'AMINCISSEMENT SEQUENTIEL**

4.1- Description.....	31
4.2- Algorithme.....	32
4.3- Algorithme de liaison.....	36
4.4- Algorithme d'amincissement.....	38
4.5- Résultats.....	40
4.6- Amélioration de l'algorithme d'Amincissement séquentiel.....	41
4.6.1- Description.....	41
4.6.2- Résultats.....	41

**CHAPITRE 5 : AMINCISSEMENT ET EPAISSISSEMENT DES MODELES  
BINAIRES.**

5.1- Nouvel algorithme d'amincissement parallèle.....	42
5.1.1- Introduction.....	42
5.1.2- Algorithme.....	42
5.1.3- Explication des quatres sous-itérations.....	43
5.1.4- Résultats.....	46
5.2- Algorithme de comparaison modifié.....	48
5.2.1- Introduction.....	48
5.2.2- Algorithme.....	48
5.2.3- Résultats.....	49

5.3- Comparaison générale.....	51
5.3.1- Algorithme de HILDITCH et le nouvel algorithme d'amincissement parallèle.....	51
5.3.2- Algorithme de comparaison et son amélioration.	51
5.3.3- L'Algorithme d'amincissement séquentiel et son amélioration.....	51
5.4- Algorithme d'épaississement.....	53
5.4.1- Introduction.....	53
5.4.2- Algorithme.....	53
5.4.3- Résultats.....	53

**CHAPITRE 6 : PRESENTATION DES RESULTATS**

Conclusion Générale.....	61
--------------------------	----

Références Bibliographiques.....	62
----------------------------------	----

**Annexes**

ANNEXE A : L'HISTOGRAMME.....	65
-------------------------------	----

ANNEXE B : PRESENTATION des PROCEDURES FORTRAN RELATIVES AUX DIFFERENTES METHODES.....	66
---	----

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

# INTRODUCTION GENERALE

## INTRODUCTION

De tous nos sens, le sens de la vue est celui qui offre la plus grande flexibilité et nous apporte la plus grande quantité d'informations sur ce qui nous entoure du fait de la multiplicité de ses dimensions : Spaciale, Energétique, Temporelle.

La transformation d'images binaires discrètes est l'opération visant à comprendre la nature d'un objet et à reconnaître sa forme.

La reconnaissance des formes peut être définie comme étant l'ensemble des techniques informatiques de représentation et de décision permettant aux machines de simuler un comportement sensible. Cette discipline permet d'augmenter la capacité de lecture à certaines machines.

Une approche particulière consiste à effectuer une transformation de voisinage, puis à employer des procédés de classification pour discriminer ou reconnaître les objets.

Notre but est d'introduire des opérations permettant l'amincissement et l'épaississement des formes.

Les Algorithmes d'amincissement des images construites à partir de diverses informations donnent une autre représentation plus facile à analyser par un observateur.

Le squelette d'une forme est la structure la plus simple possible permettant de décrire des propriétés géométriques et/ou topologiques de la forme.

Il existe un très grand nombre d'algorithmes de squelettisation d'images binaires. On peut les diviser en deux classes :

1. Les algorithmes de pelage (thinning) qui fournissent un squelette topologiquement identique à la forme (conservation de la connexité), mais ayant perdu l'information d'épaisseur de la forme, il est impossible de reconstruire la forme initiale.

2. Les algorithmes métriques pour lesquels les points du squelette sont déterminés à partir de considérations métriques et non plus topologiques; dans ce cas, on peut effectivement reconstruire la forme mais la conservation de la connexité n'est pas assurée.

Il existe des algorithmes permettant de combiner les avantages des deux méthodes précédentes: conservation de la connexité, reconstruction de la forme et squelette d'épaisseur connue à l'avance.

CHAPITRE 1

**ILLUSTRATION - DEFINITIONS  
- PROCEDURES**

## 1.1\_ INTRODUCTION :

Dans les processus de transformation d'images binaires, différentes recherches ont permis la description de plusieurs tests formels pour préserver la forme originale d'une image et pour détecter les points limites, les points finaux et les points de rupture. Avant de décrire ces procédures il est nécessaire de donner quelques illustrations et définitions de l'amincissement.

## 1.2\_ ILLUSTRATION GENERALE :

Les figures 1.1, 1.2, 1.3 et 1.4 illustrent l'opération de l'amincissement simple et de la squelettisation pour une région de forme régulière et une autre de forme irrégulière qui consiste à éliminer les points du bord qui ne sont pas des points finaux et dont l'élimination ne provoque pas des discontinuités.

### a Région de forme régulière

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

X	X	X	X	X
X	1	1	1	X
X	1	1	1	X
X	1	1	1	X
X	1	1	1	X
X	1	1	1	X
X	X	X	X	X

1	1	1
1	1	1
1	1	1
1	1	1
1	1	1

X	X	X
X	1	X
X	1	X
X	1	X
X	X	X

1
1
1

Figure 1.1 : Cas du rectangle : la valeur 1 correspond au pixel objet et X correspond aux pixels à éliminer.

1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1

X	X	X	X	X	X	X
X	1	1	1	1	1	X
X	1	1	1	1	1	X
X	1	1	1	1	1	X
X	1	1	1	1	1	X
X	1	1	1	1	1	X
X	1	1	1	1	1	X
X	X	X	X	X	X	X

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

X	X	X	X	X
X	1	1	1	X
X	1	1	1	X
X	1	1	1	X
X	1	1	1	X
X	X	X	X	X

1	1	1
1	1	1
1	1	1

X	X	X
X	1	X
X	X	X

1
---

Figure 1.2 : Cas du carré



## 1.3 \_ DEFINITIONS :

### 1.3.1 \_ LES IMAGES DISCRETES :

Afin d'être traitée par un ordinateur, une image analogique doit être numérisée, c'est-à-dire traduite sous la forme d'un tableau de nombre. Nous pouvons constater que 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.

La mémoire d'image se présente donc sous la forme d'une matrice de  $n \times p$  cases de  $K$  bits chacune (le codage correspond à la quantification de l'intensité de chaque pixel en une valeur numérique couramment appelée niveau de gris). La valeur de chaque pixel représente par exemple la transmittance moyenne (ou la densité optique) mesurée sur la surface considérée. il s'agit d'une valeur continue et il convient de la traduire par une valeur numérique généralement un nombre entier[2].

Cette valeur peut être un nombre binaire dans le cas d'une image binaire.

### 1.3.2 \_ LES IMAGES BINAIRES :

La binarisation (seuillage) d'une image présente plusieurs avantages dont l'un des plus importants est nécessairement le faible espace mémoire nécessité par une image binaire et la simplicité des opérateurs qui lui sont associés. En effet, l'image ne possédant que deux niveaux, noir et blanc, il suffit d'un bit pour coder un pixel, d'où une taille mémoire 6 fois moins élevée qu'une image codée sur 64 niveaux.

La binarisation peut être aussi considérée comme la première étape d'isolement des objets par rapport au fond.

Il est souvent utile de représenter l'image sous une autre forme que son simple affichage afin de mettre en évidence certaines régions dans le but de rendre l'image à deux niveaux de gris (binaire).

Cette forme de représentation est l'histogramme (voir AnnexeA).

### **1.3.3 \_ MODULE D'ACQUISITION :**

#### **1.3.3.1 \_ INTRODUCTION :**

La représentation d'une image sur un écran est un élément très important en imagerie. C'est en effet l'interface avec notre perception visuelle dont la subjectivité n'est plus à démontrer[1].

L'affichage d'une image monochrome en niveaux de gris peut conduire à des interprétations visuelles différentes. Ce problème s'accroît lorsque les performances du matériel d'affichage ne peuvent restituer toute l'information contenue dans l'image. Prenons un exemple : une image codée sur 256 niveaux nécessite un système capable de représenter 256 nuances sur l'écran afin d'éviter toute perte d'information, Or ce type de matériel est peu courant sur des ordinateurs personnels.

Dans le cas où nous ne disposons que d'un système qui permet d'afficher seulement deux niveaux de gris, le noir et le blanc, on aura la représentation binaire.

#### **1.3.3.2. \_ ACQUISITION :**

L'Acquisition est l'opération qui consiste à transformer un ensemble de données analogiques en un ensemble de données numériques de manière à pouvoir les traiter par ordinateur.

En vision, cette opération doit permettre de retrouver les mêmes dispositions des objets de la scène, les mêmes rapports de dimensions et de distance entre les différents objets ainsi que la même nuance de niveaux de gris.

### **1.4 \_ PROCEDURES :**

#### **1.4.1 \_ CRITERE DE SQUELETTISATION :**

Dans le processus de squelettisation du modèle binaire, l'opération d'amincissement consiste à changer successivement les points noirs en points blancs le long du contour du modèle jusqu'à ce que le modèle soit aminci.

Le modèle original peut être aminci jusqu'à son axe médian, ceci veut dire que la procédure efface les points du bord de l'objet selon un processus itératif pour trouver la position de l'axe médian de l'élément.

Simultanément le calcul de la position et la largeur est trouvé en utilisant deux tableaux, un pour placer la position et l'autre pour enregistrer la largeur.

Dans la plupart des algorithmes de squelettisation, l'opération consiste à exécuter itérativement quelque passages ou balayages sur le modèle binaire et à chaque passage des points sombres sont effacés.

A chaque balayage, le point sombre à effacer du modèle doit satisfaire les critères suivants :

- a- C'est un point du bord (c.a.d lié au contour du modèle).
- b- Ce n'est pas un point final (c.a.d que ce n'est pas un point lié aux extrémités de la ligne médiane).
- c- Ce n'est pas un point de rupture (ce qui signifie que ce n'est pas un point dont la suppression peut engendrer une déconnexion du modèle).
- d- C'est un point dont la suppression ne cause pas d'érosion excessive (n'affecte pas la forme originale de l'image).

#### 1.4.2 \_ LES OPERATIONS LOCALES :

Une opération sur une image digitalisée signifie une fonction qui transforme une matrice image  $m \times n$  donnée en une autre.

La fonction générale de ce type a( $m \times n$ ) arguments numériques (une position dans cette matrice). pour des considérations pratiques, il est préférable de travailler avec des opérations sur une image digitalisée qui peut être définie dans les termes des fonctions ayant quelques arguments considérables.

L'opération locale ou l'opération de voisinage sur une image donnée est la fonction qui définit la valeur pour chaque élément sur l'image transformée dans les limites des valeurs de l'élément correspondant et du petit ensemble de voisins dans l'image donnée. par exemple des opérations peuvent être définies en utilisant un voisinage qui consiste à donner l'élément et son 8<sup>ème</sup> voisin immédiat, une opération de ce type a seulement neuf arguments et sont sur la forme :

$$a^*_{i,j} = f(a_{i-1,j-1}, a_{i-1,j+1}, a_{i,j-1}, a_{i,j}, a_{i,j+1}, a_{i+1,j-1}, a_{i+1,j}, a_{i+1,j+1}).$$

lorsque dans un processus de transformation d'image, les opérations locales sont utilisées, l'intention est de mettre le résultats de la transformation immédiate comme des images auxiliaires; dans ce cas le résultat de la transformation n'est pas final.

Le processus de l'amincissement peut être séquentiel ou parallèle.

### 1.4.3 \_ LE PROCESSUS SEQUENTIEL :

On suppose une opération locale appliquée aux points d'une image digitalisée en une somme de séquences définies. Pour des simplifications, on suppose que les points sont traités ligne par ligne en commençant par la partie supérieure gauche.

Dans le processus séquentiel, un point est traité à n'importe quel moment; le résultat du processus est que le point à la  $(n+1)^{ème}$  itération dépend de l'ensemble des points obtenus à la  $(n)^{ème}$  itération. La séquence du processus peut être préattribuée ou bien comptée (le point suivant à considérer dépend des résultats précédents). d'Autre part la nouvelle valeur à partir de la valeur originale sera utilisée pour les points voisins suivants (qui succèdent).

Si  $g(i,j)$  et  $g'(i,j)$  marquent respectivement, les valeurs endogènes et exogènes de l'image au point  $(i,j)$  alors :

$$g(i,j) = F (g'(i-1,j-1), g'(i-1,j), g'(i-1,j+1), g'(i,j-1), g(i,j), g(i,j+1), g(i+1,j-1), g(i+1,j), g(i+1,j+1))$$

Les points  $(i-1,j-1)$ ,  $(i-1,j)$ ,  $(i-1,j+1)$  et  $(i,j-1)$  ont été traités avant les autres points restants (au moment où les points restés n'ont pas encore été procédés).

### 1.4.4 \_ LE PROCESSUS PARALLELE :

Une autre variété de procédés de transformation d'images utilisant les opérations locales est le processus parallèle. Cette approche est optimale pour des images locales.

Dans ce processus, la valeur donnée au point à la  $n^{ème}$  itération dépend de la valeur donnée au point et son 8<sup>ème</sup> voisin à la  $(n-1)^{ème}$  itération, et tous les points de l'image peuvent être traités simultanément.

Dans le cas de l'algorithme parallèle, la fonction élémentaire  $F$  est appliquée en premier ( $1_{or}$ ) à tous les points du voisinage.

$$g'(i,j) = F (g(i-1,j-1), g(i-1,j), g(i-1,j+1), g(i,j-1), g(i,j), g(i,j+1), g(i+1,j-1), g(i+1,j), g(i+1,j+1))$$

Les arguments ( $g(i-1, j-1)$ , . . . . .,  $g(i+1, j+1)$ ) sont toujours les valeurs de la matrice image originale, les nouvelles valeurs  $g'(i, j) = F(g(i-1, j-1) \dots \dots g(i+1, j+1))$  sont stockées mais pas utilisées jusqu'à ce que l'opération soit exécutée pour tout  $(i, j)$ , lorsque ensuite elles deviennent des arguments pour l'opération suivante (si besoin est).

Il est à noter que le processus séquentiel peut être plus efficace que le processus parallèle. Une transformation d'images pouvant être traitée par une série d'opérations parallèles locales peut être aussi traitée par une série d'opérations séquentielles locales, de même une transformation pouvant être traitée par une série d'opérations séquentielles locales peut être aussi traitée par une série d'opération parallèles locales.

**CHAPITRE 2**

**ALGORITHMES D'AMINCISSEMENT  
APPLIQUES AUX MODELES  
BINAIRES**

## 2.1 \_ INTRODUCTION :

L'Une des méthodes utilisées pour obtenir un squelette ou un axe médian d'une image binaire, consiste à effacer les points du contour de l'objet sans modifier les propriétés topologiques de l'image.

Il existe sur ce sujet de nombreux traités spécialisés dans le processus d'amincissement lorsqu'il est appliqué sur une image binaire tel que :

- a- Algorithmes utilisant des masques (fenêtres).
- b- Utilisation des opérations locales.
- c- La transformation de la distance.

## 2.2 \_ DEFINITION :

Soit S l'ensemble des images digitalisées et P un point de S. généralement un point de S est défini par 1 et un point n'appartenant pas à S est défini par 0.

Si P est un point de S, et les voisins de P sont notées comme dans la figure (2.1)

A	B	C
D	P	E
F	G	H

Figure 2.1 : Voisins de p .

on appelle P point limite nord si  $B=0$ ; un point limite est si  $E=0$ ; un point limite ouest si  $D=0$ ; un point limite sud si  $G=0$ , un point limite nord-est si  $B=E=0$ ; un point limite sud-est si  $E=G=0$ ; un point limite sud-ouest si  $G=D=0$ ; un point limite nord-ouest si  $D=B=0$ . Les points B,D,E et G sont appelés les 4 voisins de p (voisins orthodiagonaux) figure (2.2).

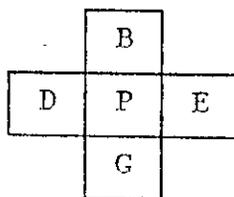


Figure 2.2 : Les quatres voisins de p.  
en ajoutant A,C,Fet H on aura les 8 voisins de P.

### 2.2.1 \_ LA CONNEXITE :

Le pas fondamental dans la description symbolique des images à partir d'un tableau de pixels ou la collection de traits primitifs est de spécifier la relation géométrique ou la connexité des pixels.

Dans l'image binaire de la figure (1.6a) l'anneau de pixels noir, divise l'image en 3 régions. Les points blancs extérieurs à l'anneau, les pixels blancs à l'intérieur de l'anneau et les pixels noirs de l'anneau lui même.

Dans chaque région les pixels sont connexes.

Une ambiguïté apparaît en considérant la figure (1.6b)

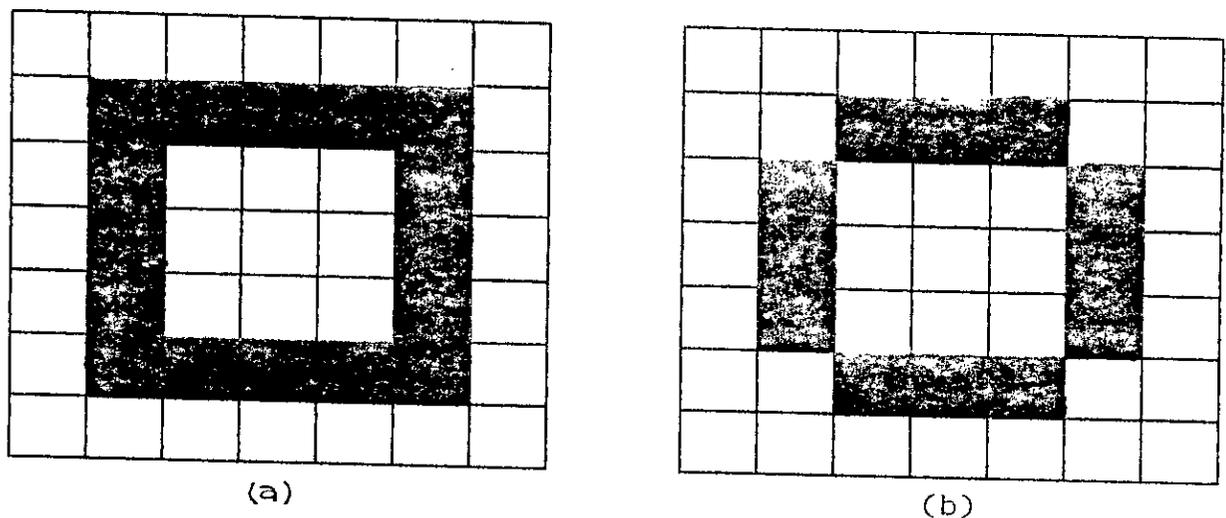


Figure 1.6 : Illustration de la connexité :  
a. Bonne figure.  
b. Mauvaise figure.

La définition de la connexité d'une image numérique n'est pas unique on peut en effet définir ce que l'on appelle la 4 connexité ou bien la 8 connexité.

#### a) LA 4 CONNEXITE :

Le pixel p et le pixel E sont connexes s'ils ont les mêmes propriétés des éléments de l'ensemble des images digitalisées S. De même, 4 connexions peuvent être établies entre le pixel p et les pixels B, D et G s'ils ont les mêmes propriétés que les éléments de l'ensemble S.[1]

## b) LA 8 CONNEXITE :

Le pixel  $p$  et un de ses voisins de diagonale,  $P$  et  $C$  par exemple, sont connexes s'ils ont les mêmes propriétés des éléments de l'ensemble des images digitalisées  $S$ .

On note que le pixel  $p$  a les mêmes propriétés que les éléments de l'ensemble  $S$  si ce pixel est un point de  $S$  avec la valeur 1.

Donc  $S$  est appelé 4-connexions si pour deux pixels  $p$  et  $Q$  de  $S$  il existe une séquence de points.

$$P = P_0, P_1, P_2, \dots, P_n = Q$$

de  $S$  tel que  $P_i$  est le 4<sup>ème</sup> voisin de  $P_{i-1}$ ,  $1 < i < n$ .

de même  $S$  est appelé 8-connexions si pour deux pixels  $P$  et  $Q$  de  $S$  il existe une séquence de points.

$$P = P_0, P_1, P_2, \dots, P_n = Q$$

de  $S$  tel que  $P_i$  est le 8<sup>ème</sup> voisin de  $P_{i-1}$ ,  $1 < i < n$ . [1].

### **2.2.2\_ LE POINT FINAL - LE POINT ISOLE :**

On appelle  $P$  le 4<sup>ème</sup> point final si un de ces quatre voisins est un (1) et le 8<sup>ème</sup> point final si exactement un de ces huit voisins est un (1).

Par exemple ,

0	0	0
0	P	1
1	0	1

$P$  est 4<sup>ème</sup> point final (mais pas 8<sup>ème</sup> point final).

On appelle  $P$  point isolé d'ordre 4 si aucun de ses quatre voisins n'est égal à 1 et un point isolé d'ordre 8 si aucun de ses huit voisins n'est égal à 1.

### 2.2.3 \_ LE POINT DE RUPTURE :

Le pixel P est un point de rupture s'il a la valeur 1 et la fenêtre 3 - par - 3 centrée sur ce point doit être pareil à l'une des six fenêtres suivantes notées S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, S<sub>4</sub>, S<sub>5</sub> et S<sub>6</sub>.

*	*	*
0	1	0
*	*	*

S1

*	0	*
*	1	*
*	0	*

S2

*	*	*
*	1	0
*	0	1

S3

*	*	*
0	1	*
1	0	*

S4

1	0	*
0	1	*
*	*	*

S5

*	0	1
*	1	0
*	*	*

S6

Figure 2.3 : Les six fenêtres S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, S<sub>4</sub>, S<sub>5</sub> et S<sub>6</sub>,

\* Peut prendre la valeur 0 ou 1.

**CHAPITRE 3**

**ETABLISSEMENT DES ALGORITHMES  
DE COMPARAISON ET DES  
OPERATIONS LOCALES**

### 3.1- (ALGORITHME DE COMPARAISON) :

#### 3.1.1\_ INTRODUCTION :

Le concept de comparaison par fenêtres est largement utilisé dans diverses applications en raison de sa simplicité. Pour l'image digitale, la fenêtre (masque) est un tableau spécifié pour détecter les propriétés des régions invariantes. Une des méthodes fondamentales de reconnaissance d'un objet dans une image est la comparaison assortie (en accord) ou un modèle (repère) est comparé à tous les objets inconnus dans l'image.

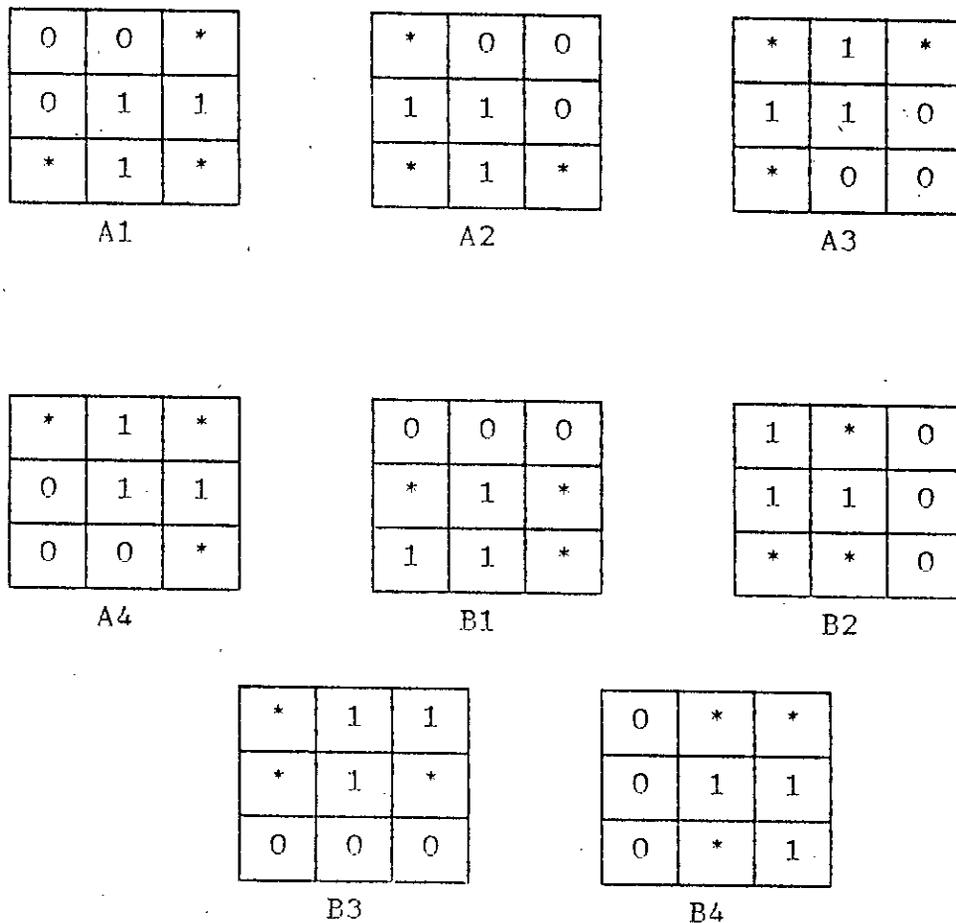
Si la fenêtre de comparaison sera fermée, ceci veut dire que l'objet inconnu est désigné comme étant fenêtre objet (c.a.d l'objet à reconnaître dans l'image sera l'objet lui même une fois accord des fenêtres).

#### 3.1.2\_ L'ALGORITHME D'ARCELLI :

L'Algorithme d'ARCELLI utilise l'ensemble de huit (08) fenêtres (masques) notées par  $A_1, A_2, A_3, A_4, B_1, B_2, B_3$ , et  $B_4$  montrées dans la figure (3.1).

L'Algorithme de comparaison est un algorithme d'amincissement où les pixels de l'image binaire sont éliminés en commençant du haut coin gauche au bas coin droit de l'image et en effaçant à chaque fois les points du bord (du contour) de l'objet à traiter jusqu'à l'obtention d'un modèle aminci dont la largeur est de un pixel. [22]

Les pixels sont éliminés après comparaison de chaque pixel de valeur 1 et ses voisins dans l'image avec l'ensemble des fenêtres.



**Figure 3.1 :** "ensemble de fenêtres"; la valeur 1 représente la valeur de l'objet; la valeur 0 représente la valeur de l'arrière plan, et \* valeur quelconque (i.e peut prendre la valeur 1 ou 0).

Dans ces fenêtres, les zéros doivent être en accord avec les valeurs 0 des pixels de l'image, les uns doivent être en accord avec les valeurs 1 des pixels et l'astérisque peut prendre la valeur 1 ou la valeur 0 des pixels dans l'image. On commence par la fenêtre A<sub>1</sub> qui sera comparée à tous les pixels de valeur 1 et leurs voisins dans l'image. Si l'accord est obtenu, le pixel central correspondant de l'image sera effacé (changé à la valeur 0). Après cette procédure avec la fenêtre A<sub>1</sub> le processus est répété en utilisant la fenêtre B<sub>1</sub>, ensuite avec A<sub>2</sub>, B<sub>2</sub>, A<sub>3</sub>, B<sub>3</sub>, A<sub>4</sub> et B<sub>4</sub> dans cet ordre en formant un cycle complet. Le processus s'arrête lorsqu'il n'y a aucun pixel à éliminer durant la procédure du cycle complet. L'organigramme de l'algorithme de comparaison est illustré à la figure (3.2).

### 3.1.3\_ RESULTATS :

En utilisant l'algorithme de comparaison[22-23] où huit fenêtres (masques)  $A_1, A_2, A_3, A_4, B_1, B_2, B_3, B_4$  et seulement une image sont utilisés dans le processus; on obtient un squelette non centré et avec la présence de cerifs dans certaines parties de l'objet (squelette).

Le nom du programme relatif à cet algorithme : TEMP1.FOR (voir Annexe B).

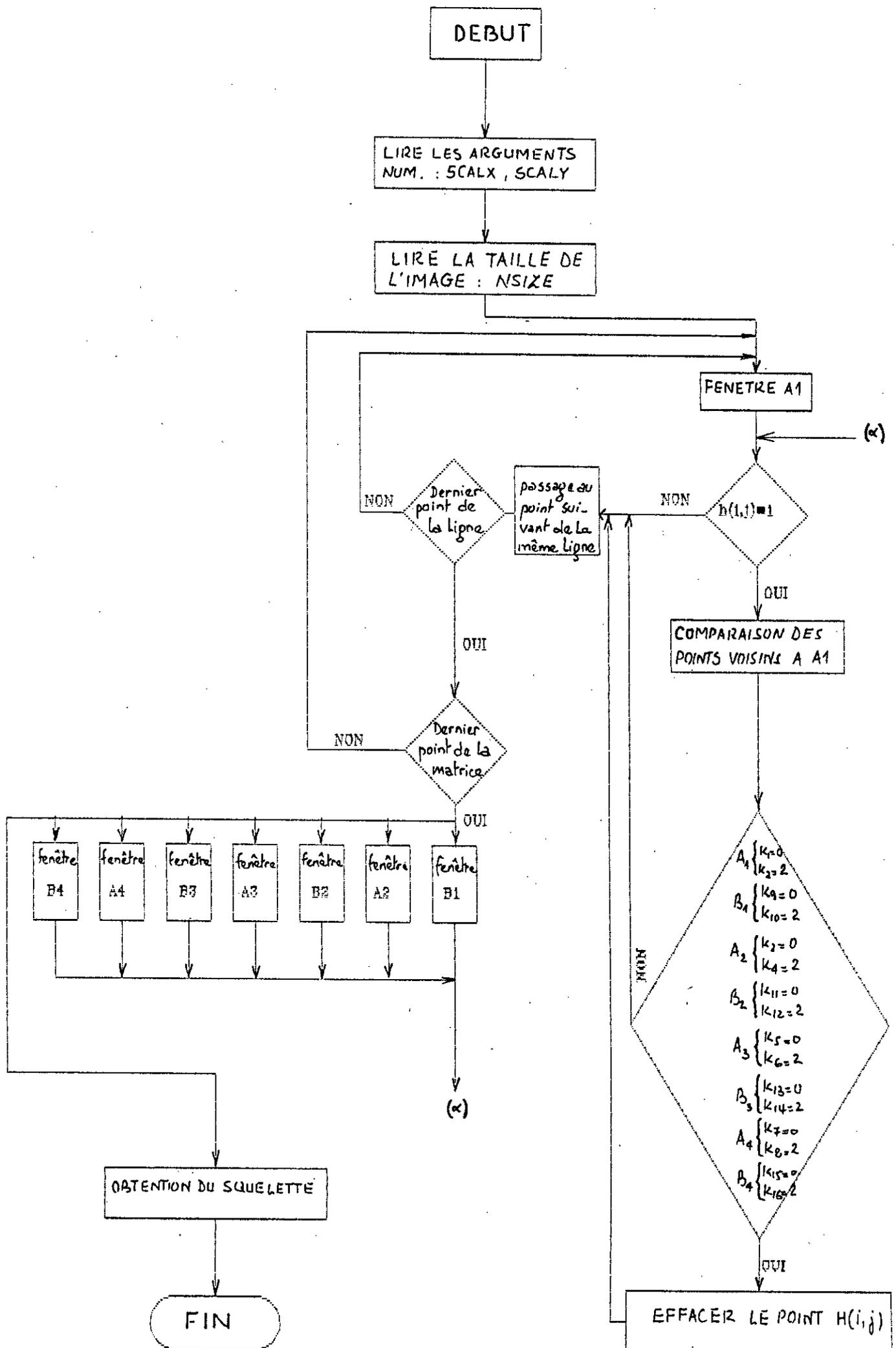


Figure 3.2 : organigramme de l'algorithme de comparaison.

Dans cet organigramme (fig 3.2) la taille de l'image  $n \text{ size}$  est le nombre de pixels dans l'image binaire; cette dernière est une matrice de caractères  $I$  ( $n \text{ size}, n \text{ size}$ ) dont les 1 représentent l'objet à traiter et les 0 constituent l'arrière-plan.

les  $K_i$  sont les paramètres des huit fenêtres de comparaisons tel que :

$$K_1 = A + B + D \text{ (voir figure 2.1)}$$

$$K_2 = E + G$$

$$K_3 = B + C + D$$

$$K_4 = D + G$$

$$K_5 = E + H + G$$

$$K_6 = D + B$$

$$K_7 = D + F + G$$

$$K_8 = B + E$$

$$K_9 = A + B + C$$

$$K_{10} = F + G$$

$$K_{11} = C + E + H$$

$$K_{12} = A + D$$

$$K_{13} = F + G + H$$

$$K_{14} = B + C$$

$$K_{15} = A + D + F$$

$$K_{16} = E + H$$

## 3.2 ALGORITHME D'AMINCISSEMENT PARALLELE :

### 3.2.1. INTRODUCTION :

Dans le processus d'amincissement, des recherches particulières ont permis la description de plusieurs tests pour préserver la forme de l'image originale et pour détecter les points finaux, les points limites et les points de rupture.

### 3.2.2 L'ALGORITHME DE NACCACHE :

L'Algorithme de NACCACHE est basé sur le critère de squelettisation précédent.

Pour cet algorithme on définit[34] :

1.  $B(P_1)$  : le nombre de voisins non nuls de  $P_1$

$$B(P_1) = P_2 + P_3 + P_4 + P_5 + P_6 + P_7 + P_8 + P_9$$

2. Un point limite comme étant un point noir avec au moins un des quatres voisins blanc.

3. Un point final comme étant un point noir avec au plus un des huit voisins noir.

4.  $X(P_1)$  : définit par :

$$X(P_1) = \sum_{i=2}^5 b_i$$

$$\left[ \begin{array}{l} \text{ou} \\ b_i = 1 \quad \text{si } (P(2i-2) \text{ est blanc et } (P(2i-1) \text{ ou } (P(2i) \text{ est} \\ b_i = 0 \quad \text{ailleurs} \quad \text{noir)} \end{array} \right.$$

c.a.d d'une autre façon on a :

$$X(P_1) = b_2 + b_3 + b_4 + b_5 \quad \text{tel que :}$$

$$b_2 = \begin{cases} 1 & \text{si } (P(2) = 0) \text{ et} \\ & (P(3) = 1 \text{ ou } P(4) = 1) \\ 0 & \text{sinon} \end{cases}$$

$$b_3 = \begin{cases} 1 & \text{si } (P(4) = 0) \text{ et} \\ & (P(5) = 1 \text{ ou } P(6) = 1) \\ 0 & \text{sinon} \end{cases}$$

$$b_4 = \begin{cases} 1 & \text{si } (P(6) = 0) \text{ et} \\ & (P(7) = 1 \text{ ou } P(8) = 1) \\ 0 & \text{sinon} \end{cases}$$

$$b_5 = \begin{cases} 1 & \text{si } (P(8) = 0) \text{ et} \\ & (P(9) = 1) \\ 0 & \text{sinon} \end{cases}$$

Dans une passe donnée, le modèle proposé est balayé de gauche à droite et du haut vers le bas.

Le point  $P_1$  est repéré si les 6 tests suivants sont vérifiés.

1.  $P_2 + P_4 + P_6 + P_8 \leq 3$  .... ( $P_1$  est un point limite).
2.  $B(P_1) \geq 2$  ..... ( $P_1$  n'est pas un point final).
3.  $N(P_1) \geq 1$  .....  $N(P_1)$  est le nombre de points non repérés (maintenus).
4.  $X(P_1) = 1$  ..... ( $P_1$  n'est pas un point de rupture).

Ce test a été utilisé dans le premier algorithme présenté précédemment lors de la construction des huit fenêtres de comparaison. Par exemple la fenêtre  $A_1$  montré dans la figure (3.3) a été construite tel que  $X(P_1) = 1$ . en effet

$$X(P_1) = b_2 + b_3 + b_4 + b_5 \quad \text{avec}$$

$$. \quad b_2 = 1 \text{ car } P(2) = 0 \text{ et } P(4) = 1$$

$$. \quad b_3 = 0 \text{ car } P(4) = 1$$

$$. \quad b_4 = 0 \text{ car } P(6) = 1$$

.  $b_5 = 0$  car  $P(8) = 0$  et  $P(9) = 0$ .

On a bien  $X(P_1) = 1 \implies P_1$  n'est pas un point de rupture d'ou on voit bien que la fenêtre  $A_1$  centré sur le point  $P_1$  vérifie bien l'un des critères de squelettisation.

0 P(9)	0 P(2)	* P(3)
0 P(8)	1	1 P(4)
* P(7)	1 P(6)	* P(5)

Figure 3.3 : la fenêtre  $A_1$  de l'algorithme de comparaison.

5.  $P_4$  est non repéré ou  $X_4(P_1) = 1$  ce test préserve la forme de l'image.

6.  $P_6$  est maintenu (non repéré) ou  $X_6(P_1) = 1$  ce test préserve la forme de l'image comme le test (5).

A la fin du balayage tous les points repérés sont effacés.

### 3.2.3\_\_ ANALYSE DE L'ALGORITHME :

Les six opérations données précédemment peuvent être expliquées par les étapes suivantes[34] :

Pour éliminer le point P, on doit vérifier que P n'est pas un point final, ni un point de rupture, et que sa suppression ne cause pas d'érosion. Pour chaque point P on doit comparer le voisinage de P avec les quatres fenêtres de la figure (3.4).

Si le voisinage de P sera en accord avec l'une des quatres fenêtres, alors le point P n'est pas repéré. Notons que les points repérés par X et Y dans les fenêtres n'influent pas sur les opérations à traiter et ne portent aucune condition (i.e le blanc ou le noir pour le X et le Y ne change pas le processus). On examine maintenant les quatres fenêtres une par une, pour justifier pourquoi ces fenêtres permettent le test du point P pour le point de rupture, le point final et l'érosion.

Si le voisinage de P sera en accord avec l'une des fenêtres (a), (b) ou (c) de la figure (3.4), deux situations se présentent :

1. Si tous les X sont blancs, alors P est un point final.
2. Si au moins un des X est un point noir, alors P est un point de rupture. Dans ces deux cas, P ne doit pas être repéré.

Examinons maintenant la fenêtre (d) de la figure (3.4). si au moins un X et au moins un Y est noir, alors P devient un point de rupture et ne doit pas être repéré. En facilitant l'analyse, on suppose qu'en premier lieu tous les X sont blancs, alors il y'a huit configurations possibles comme le montre la figure (3.5). dans les configurations W1, W2, et W3 le pixel P est un point final, et la configuration W4 fait que le point P soit un point de rupture; dans les configurations W5, W6, et W7 le point P ne doit pas être enlevé (trouvé expérimentalement par NACCACHE). dans les configurations W7 et W8, le point P est en fait un bruit. Le bruit doit être enlevé à priori des modèles traités, d'où les configurations W7 et W8 peuvent ne pas existé au début du processus de squelettisation. d'autre part, un point qui porte une information sur la forme du modèle dans l'image ne doit pas être effacé.

Par symétrie, on prolonge notre analyse dans le cas, dans la fenêtre (d), où tous les Y sont des point blancs et les X prennent des valeurs variables de blancs et de noir, d'où pour la fenêtre (d), les points X et Y ont des valeurs qui ne changent pas le processus.

1	0	X
0	P	X
X	X	X

(A)

X	X	X
0	P	X
1	0	X

(B)

X	0	0
0	P	1
X	0	0

(C)

X	X	X
0	P	0
Y	Y	Y

(D)

**Figure 3.4** : Si le voisinage du point noir ressemble à l'une des quatre fenêtres, le point P n'est pas repéré; X et Y ont des valeurs quelconques.

Les huit fenêtres ou configurations possible pouvant être dérivées de la fenêtre (d) de la figure (3.6) sont montrées dans la figure (3.5).

0	0	0
0	P	0
1	0	0

W1

0	0	0
0	P	0
0	1	0

W2

0	0	0
0	P	0
0	0	1

W3

0	0	0
0	P	0
1	0	1

W4

0	0	0
0	P	0
0	1	1

W5

0	0	0
0	P	0
1	1	0

W6

0	0	0
0	P	0
1	1	1

W7

0	0	0
0	P	0
0	0	0

W8

**Figure 3.5** : huit fenêtres dérivées de la fenêtre (d).

### 3.2.4\_ RESULTATS :

L'Algorithme d'amincissement parallèle donné par N.NACCACHE [34] conserve bien la forme de l'image binaire originale mais a les deux problèmes suivants :

1. Cet algorithme laisse des pixels étranges (branches) dans le squelette.
2. Le squelette final n'est pas connexe.

Le nom du programme : ALH.FOR (voir Annexe B).

### 3.3\_ LA VERSION SIMPLIFIEE DE HILDITCH :

#### 3.3.1\_ DESCRIPTION :

La meilleur squelettisation connue est celle de l'algorithme de HILDITCH, HILDITCH décrit en détail le critère à satisfaire avant que le point noir du modèle ne soit supprimé[15].

L'image digitale est définie par une matrice A carrée, dont les arguments numériques  $a_{i,j}$  sont soit 1 soit 0. On suppose que l'objet dans l'image est représenté par les éléments de valeur 1.

Généralement le processus de transformation est itérativement appliqué à la matrice A, où tous les points transformés dépendent d'un petit ensemble de points voisins (c.a.d fonction booléenne).

Les points voisins du point  $(i,j)$  sont :

$(i,j+1)$ ,  $(i-1,j+1)$ ,  $(i-1,j)$ ,  $(i-1,j-1)$ ,  $(i,j-1)$ ,  $(i+1,j-1)$ ,  $(i+1,j)$ ,  $(i+1,j+1)$  et le point  $(i,j)$  lui même comme il est illustré sur la figure (3.6).

$P_9$ $(i-1, j-1)$	$P_2$ $(i-1, j)$	$P_3$ $(i-1, j+1)$
$P_8$ $(i, j-1)$	$P_1$ $(i, j)$	$P_4$ $(i, j+1)$
$P_7$ $(i+1, j-1)$	$P_6$ $(i+1, j)$	$P_5$ $(i+1, j+1)$

Figure 3.6: Description des neuf pixels dans la fenêtre 3-pas-3 .

### 3.3.2\_ ANALYSE DE L'ALGORITHME :

La première méthode possible pour l'extraction de la ligne médiane (i.e le squelette) de la figure, consiste à enlever tous les points du contour de la figure à l'exception des uns appartenant au squelette. Les huit points  $P_2$ ,  $P_3$ ,  $P_4$ ,  $P_5$ ,  $P_6$ ,  $P_7$ ,  $P_8$  et  $P_9$  sont connus comme étant les 8- voisins de  $P_1$ . Le contour du point  $P_1$  est effacé du modèle binaire si les tests suivant sont satisfaits :

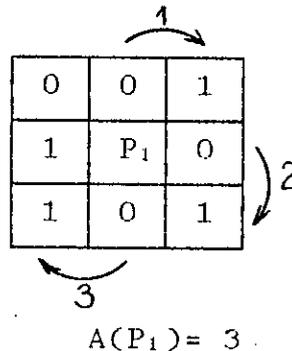
1.  $2 \leq B(P_1) \leq 7$
2.  $A(P_1) = 1$
3.  $P_2 * P_4 * P_6 = 0$
4.  $P_2 * P_4 * P_8 = 0$

où :

$A(P_1)$  est le nombre de 0-1 variation dans l'ordre  $P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9$  et  $P_2$  qui sont les huit voisins de  $P_1$ .

$B(P_1) = P_2 + P_3 + P_4 + P_5 + P_6 + P_7 + P_8 + P_9$ .

Si une de ces conditions n'est pas satisfaite, alors le point  $P_1$  n'est pas effacé de l'image. ceci peut être illustré par l'exemple de la figure (3.7) où  $A(P_1) = 3$ .



**Figure 3.7 :** Le nombre de 0-1 variation dans l'ordre  $P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9$  et  $P_2$ .

### 3.3.3\_ RESULTATS :

La version simplifiée de HILDITCH est décrite comme suit[15]:

a)- PROCEDURE (a) :

Le point  $P_1$  est éliminé (changé à un pixel de valeur égal à 0) du modèle binaire traité si :

- a.  $2 \leq B(P_1) \leq 7$
- b.  $A(P_1) = 1$
- c.  $P_2 * P_4 * P_6 = 0$
- d.  $P_2 * P_4 * P_8 = 0$

Sont satisfaites. Cet algorithme ne préserve pas exactement la forme originale du modèle et produit parfois des pixels étranges (branches) sur le squelette final.

Le nom du programme : ALGSUB10.FOR (Voir Annexe B).

**b)- PROCEDURE (b) :**

Le point  $P_i$  est repéré (étiqueté avec une autre valeur que 0 ou 1).

Dans chaque balayage, et à la fin de chaque passe, tous les points repérés sont effacés.

Dans ce cas l'algorithme produit les résultats suivants :

1. On obtient un squelette mais pas au milieu (c.a.d non centré).
2. La forme de l'image originale n'est pas toujours préservé ainsi que la présence de pixels étranges (branches).

Le nom du programme : ALCSUB1.for (voir Annexe B).

**CHAPITRE 4**

**ALGORITHME D'AMINCISSEMENT  
SEQUENTIEL**

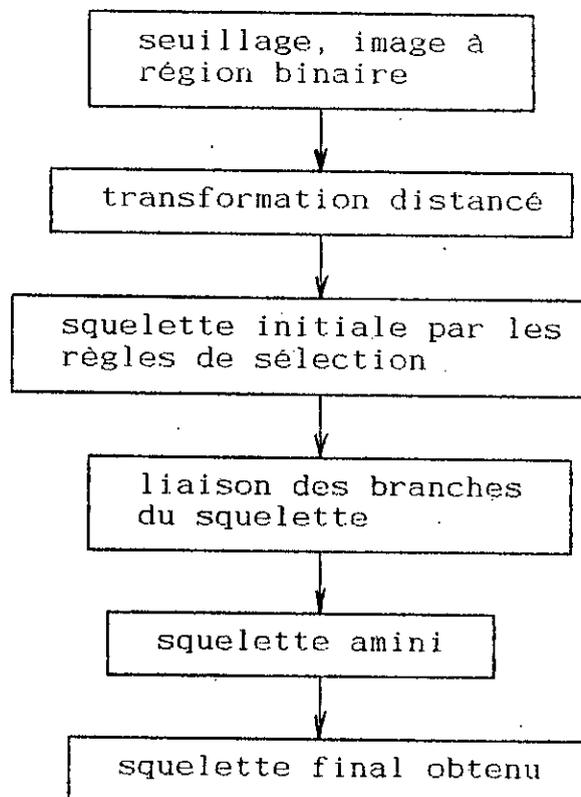
## ALGORITHME D'AMINCISSEMENT SEQUENTIEL

### 4.1- DESCRIPTION :

l'Algorithme d'amincissement séquentiel consiste à faire :

1. l'Application séquentielle de la transformation de la distance (DT).
2. Algorithme de liaison; et
3. Algorithme d'amincissement.

l'Organigramme de la figure (4.1) montre le processus d'analyse de l'image pour l'extraction de la forme ainsi que l'information sur la position de l'image binaire, suivi de la transformation distancé (DT). Le squelette initial est obtenu par application de la première ou la deuxième équation de la règle de sélection, l'algorithme de liaison est appliqué immédiatement après la sélection de la règle pour obtenir un squelette connexe, et puisque le squelette obtenu ici n'a pas dans la plupart des cas un épaisseur de 1 pixel, une procédure d'amincissement est nécessaire pour obtenir le squelette final[18-19].



**Figure 4.1:** Organigramme de l'algorithme d'amincissement séquentiel.

## 4.2\_ ALGORITHME :

### 4.2.1\_ LA TRANSFORMEE DE LA DISTANCE :

La transformée de la distance comprend la DT et l'ensemble des règles de sélection pour extraire des éléments du squelette.

### 4.2.2\_ LA TRANSFORMATION DE LA DISTANCE (DT) :

La transformation de la distance est utilisée dans la détermination du squelette, ici le temps du processus est très rapide.

Soit une image digitalisée dont les éléments ont seulement les valeurs 0 ou 1, on fait la construction de la DT de l'image dont chaque élément a une valeur entière égale à la distance le séparant des 0, (l'ensemble des 0 est non vide). En particulier, les 0 restent inchangés car ils sont à zéro distance d'eux même, les 1 qui sont les voisins horizontaux ou verticaux des 0 restent aussi inchangés, les 1 qui sont des voisins horizontaux ou verticaux de tout 1 deviennent 2 et le processus continu.

La transformation de la distance utilisée peut être illustré en utilisant la méthode suivante :

1. Au premier balayage l'image est parcourue par ligne de haut en bas et de gauche à droite. Tout pixel de valeur 1 dans l'image sera affecté par une nouvelle valeur tel que :

$$D(\text{ligne}, \text{colonne}) = \min(D(\text{ligne}-1, \text{colonne}), D(\text{ligne}, \text{colonne}-1)) + 1.$$

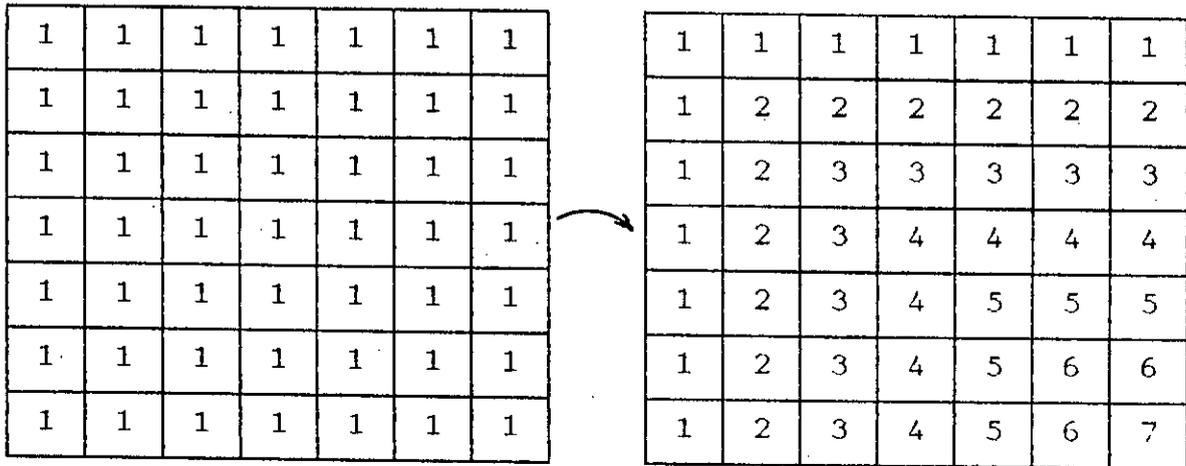
C'est le calcul de la distance de chaque pixel de gauche au coté le plus élevé (loin) de la ligne.

2. Au deuxième balayage, l'image est parcourue par ligne de bas en haut et de droite à gauche. Chaque pixel de valeur non nulle dans l'image sera affecté par une nouvelle valeur tel que :

$$D(\text{ligne}, \text{colonne}) = \min(D(\text{ligne}+1, \text{colonne})+1, D(\text{ligne}, \text{colonne}+1)+1, D(\text{ligne}, \text{colonne})) .$$

Comme exemple à cette étape la figure (4.2) montre la transformation effectuée en deux balayage :

1- Le premier balayage :



2- Le deuxième balayage :

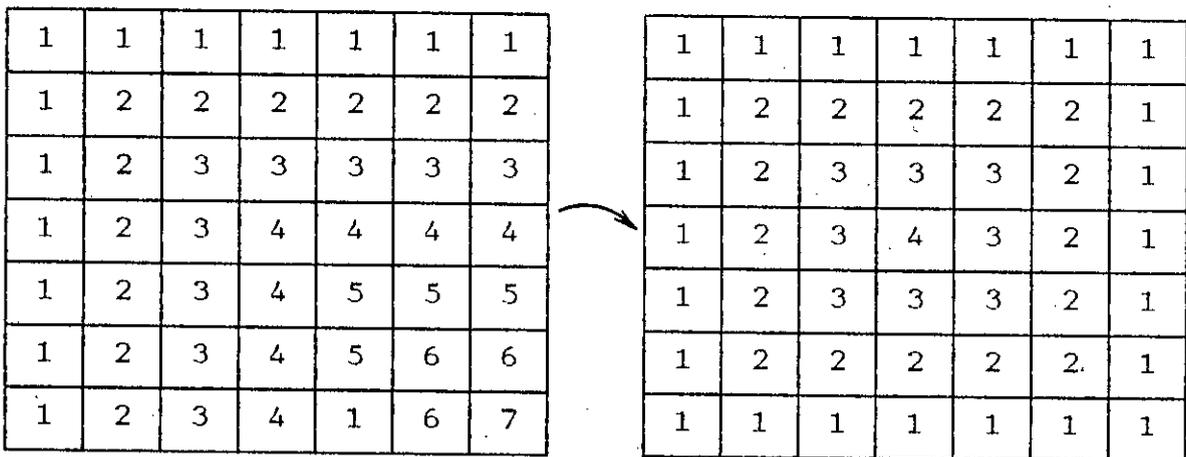


Figure 4.2 : Exemple de l'utilisation de la DT donnée précédemment.

La plus simple définition de la DT par rapport à ce qui vient d'être expliqué précédemment est donnée par :

Soit S un sous ensemble de l'image digitale, un point de S est représenté par la valeur 1 et un autre n'appartenant pas à S est représenté par 0.

Pour tout point P de S,  $d(P)$  est la distance du point P au plus proche point n'appartenant pas à S.

La distance entre deux points  $(x,y)$  et  $(u,v)$  peut être obtenue en prenant la distance minimale entre  $(x,y)$  et  $(u,v)$  dans les quatre directions (nord, sud, est et ouest).

On note que  $(x,y)$  est un point de S (c.a.d un pixel de valeur égale à 1) et  $(u,v)$  n'est pas un point de S (c.a.d un pixel de valeur égale à 0).

En utilisant cette méthode on obtient le même résultat que précédemment (voir figure (4.2)).

#### 4.2.3\_ LES REGLES DE SELECTION :

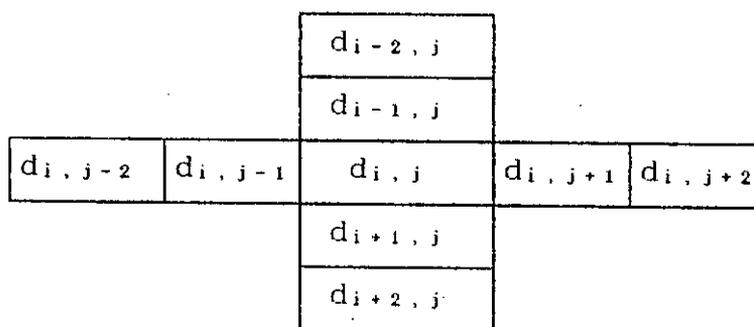
Le but des règles de sélection est d'engendrer un squelette initial et de sélectionner des pixels comme des éléments squelette. Il y'a deux façons de procéder par les règles de sélection.

La première sélectionne le squelette en utilisant les 4 voisins orthodiagonaux : un pixel de chaque côté; en utilisant la 1ere équation on a :

l'équation (a) peut être décrite par :

Si  $(D(i,j)+1 > D(i,j-1), D(i,j)+1 > D(i-1,j),$   
 $D(i,j)+1 > D(i,j+1) \text{ et } D(i,j)+1 > D(i+1,j)$   
 alors  $D2(i,j) = D(i,j)$   
 sinon  $D2(i,j) = 0$

La deuxième sélectionne le squelette en utilisant les 4 voisins prolongés : Deux pixels de chaque côté respectant l'équation (b) comme il est montré à la figure (4.3).



**Figure 4.3 :** Voisins utilisés dans les règles de sélection.

l'Equation (b) peut être décrite par :

Si  $(D(i,j)-D(i,j-2)+D(i,j)-D(i-2,j)+D(i,j)-D(i+2,j)+D(i,j) - D(i,j+2) > 2)$

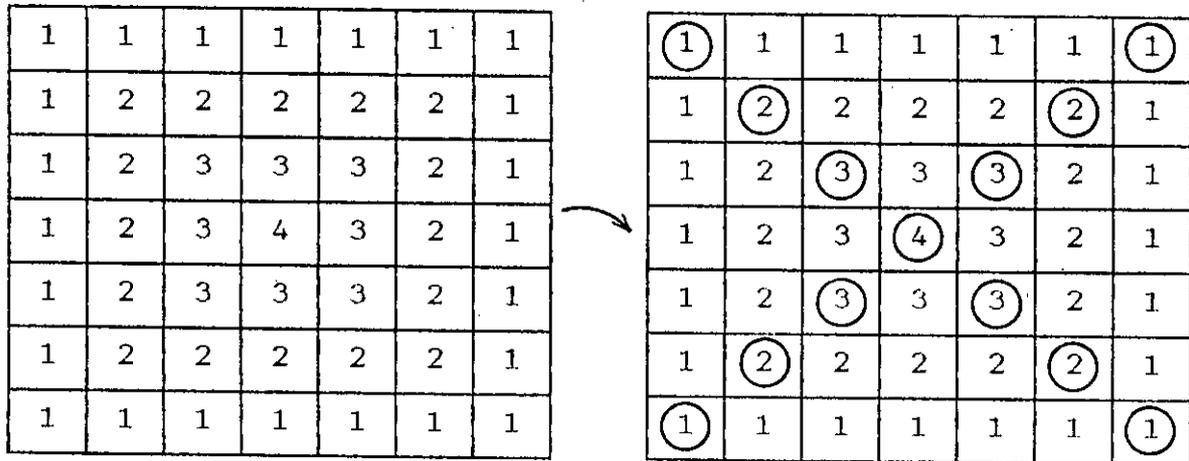
alors  $D2(i,j) = D(i,j)$

sinon  $D2(i,j) = 0$

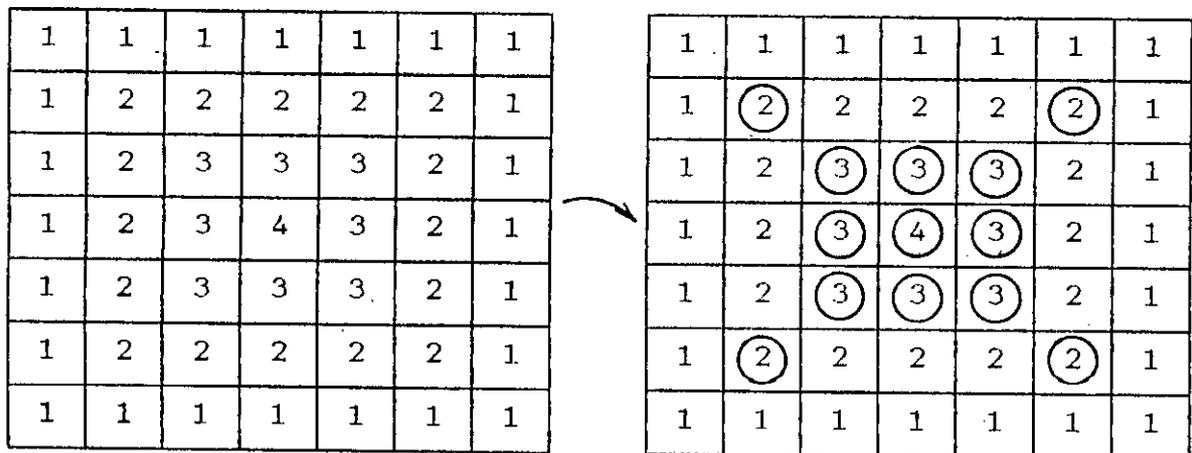
Notons que l'utilisation de la deuxième équation permet de produire un squelette qui est plus connexe que le squelette obtenu en utilisant la première équation.

$D(i,j)$  représente la DT et  $D2(i,j)$  représente le squelette initial obtenu en utilisant les règles de sélection.

Un exemple d'utilisation des équations (a) et (b) est montré à la figure (4.4).



(a)



(b)

**Figure 4.4 :** a\_ Squelette obtenu en utilisant l'équation (a)  
 b\_ squelette obtenu en utilisant l'équation (b)

### **4.3 \_ ALGORITHME DE LIAISON :**

L'Algorithme de liaison est appliqué immédiatement après les règles de sélection pour produire un squelette connexe.

L'Algorithme de liaison permet de remplir les vides produits par les règles de sélection, d'où squelette connexe peut être obtenu.

L'Algorithme peut être divisé en deux étapes :

1. Test de linéarité.
2. Processus de liaison

Dans un but de minimisation du temps de processus, le test de linéarité est introduit pour trouver une possibilité de connexion par segment à l'intérieur d'une fenêtre 3-par-3 centrée au pixel courant. Si un segment linéaire peut être détecté dans une direction horizontale, verticale, ou bien dans une direction diagonale à l'intérieur d'une fenêtre 3-par-3, le processus de liaison est évité, et ainsi le temps demandé pour accomplir le processus de liaison est gagné. Si le test de linéarité ne peut détecter quelque segments de connexion à l'intérieur de la fenêtre 3-par-3, les pixels sont testés par extension avec une fenêtre 7-par-4 pour le processus de liaison.

#### **4.3.1 \_ TEST DE LINEARITE :**

Le test de linéarité est procédé par une fenêtre 3-par-3 centrée sur le point courant  $P_{i,j}$ . Il est seulement exécuté lorsque la valeur de la DT de chaque point image est plus grande que zéro, c.a.d lorsque le point image appartient au squelette. Le segment de connexion squelette est défini par :

- (i) soient trois éléments, dans la même ligne ou colonne, ou deux axes diagonaux quelconque de la fenêtre.
- (ii) Toutes les valeurs de la DT des trois éléments doivent être supérieur à zéro.

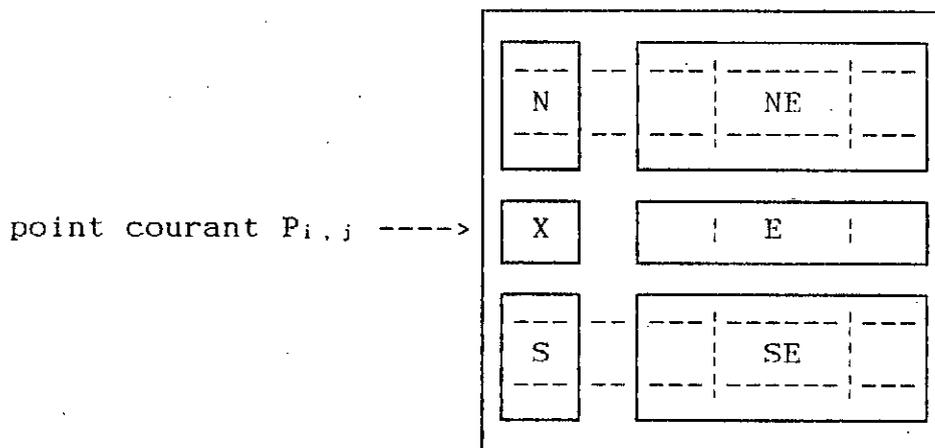
#### **4.3.2 \_ LE PROCESSUS DE LIAISON :**

Le processus de liaison est procédé par une fenêtre 7-par-4 comme il est illustré à la figure (4.5). La fenêtre est utilisée de la même manière que la fenêtre 3-par-3. Le pixel courant,  $P_{i,j}$  est localisé au milieu de l'élément de la colonne la plus à gauche de la fenêtre 7-par-4.

La surface de liaison est subdivisée en cinq régions (voir fig(4.5)) : nord(N); nord-est (NE); est (E); sud-est (SE); et sud (S). Les opérations de liaison sont exécutées séquentiellement dans le sens des aiguilles d'une montre à partir de la région Nord à la région Sud. Les régions NE et SE sont des fenêtre 3-par-3; le N et le S sont des fenêtres 3-par-1; et la région E et une fenêtre 1-par-3.

Dans chaque région, si l'élément Correspond au pixel courant (ou bien un de ces 8-voisins) est un élément squelette, les opérations de liaisons sont évitées et procédées à la région suivante par séquence. Dans les régions N, NE et E, les opérations de liaison permettent de transformer deux composants connexes en un seul composant connexe, c.a.d faire la liaison des deux composants; mais dans les régions SE et S, elles permettent de réduire le vide entre deux composants connexes de deux pixels en un seul.

Lorsque le composant connexe est obtenu ou bien le vide a été réduit dans la région, le processus de liaison est terminé pour ce pixel particulier, et la fenêtre de liaison passe au pixel voisin.



**Figure 4.5 :** Fenêtre 7-par-4 pour la liaison squelette.

Le squelette résultant est connexe, mais l'épaisseur de ce dernier n'est pas en général égale à un pixel, par conséquent une procédure d'amincissement est nécessaire pour obtenir le squelette (i.e squelette d'épaisseur d'un pixel).

#### 4.4- ALGORITHME D'AMINCISSEMENT :

Tout les éléments squelette sélectionnés sont actuellement en valeur de DT. ces éléments sont convertis à 1, les autres éléments, non squelette, de même ceux dont les valeurs de la DT sont supérieurs à 0, sont convertis à 0.

L'Algorithme présenté permet de déterminer par des opérations successives si le pixel courant  $P_0$  peut être enlevé ou non.

La fenêtre est déplacé du haut en bas et de gauche à droite de l'image. Si la connexité de tout les pixels voisins du squelette dans la fenêtre est préservée après avoir effacé le pixel central de la fenêtre, le pixel central est effacé d'une façon permanente (changé en zéro). d'autre part, si la fenêtre ne contient qu'un seul pixel voisin, le pixel central n'est pas enlevé.

Si elle ne contient aucun pixel voisin, le pixel central (un pixel isolé) est enlevé.

L'Algorithme d'amincissement proposé consiste à effectuer les six opérations suivantes :

##### 4.4.1\_ STOCKAGE :

Tous les voisins  $P_i$  pour  $1 \leq i \leq 8$  sont représentés respectivement par 1 ou bien 0 dans un emplacement mémoire T; ( $T_8$  est le bit le plus signifiant dans la position la plus à gauche), voir figure (4.6).

##### 4.4.2\_ MASQUE :

Les bits  $T_2$ ,  $T_4$ ,  $T_6$  et  $T_8$  du byte T correspondant aux quatres coins de la fenêtre sont masqués (changés en zéro) et le résultat est stocké dans  $T_M$ .

##### 4.4.3\_ DECALAGE :

Le contenu de  $T_M$  est décalé à gauche par un bit (avec le dernier reste), et le résultat est représenté par Q.

##### 4.4.4\_ OU. "OR" :

Les contenus de T et Q sont liés ensemble par l'opération "ou" et stockés dans  $T_R$ .

#### 4.4.5\_ COMPTAGE :

Les bits dans chaque paire de positions consécutives (y compris le dernier reste (carry)) dans  $T_R$  sont comparés. Pour tout changement de 1 à 0 dans la paire, le compteur de changement  $N_{1 \rightarrow 0}(T_R)$  est incrémenté, le nombre totale de 1 dans  $T_R$  ( $N_1(T_R)$ ) est aussi compté.

#### 4.4.6\_ DECISION :

Si le compteur de changement de pas 5,  $N_{1 \rightarrow 0}(T_R)$  est plus grand que 1, le point courant  $P_0$  n'est pas effacé. Si  $N_{1 \rightarrow 0}(T_R)$  contient 1, et le nombre totale de 1 dans  $T_R$  ( $N_1(T_R)$ ) est aussi 1 le pixel courant n'est pas enlevé; dans le cas contraire le pixel courant est effacé. La figure (4.6) présente un voisinage 3-par-3 avec l'étiquetage des pixels.

$P_8$	$P_1$	$P_2$
$P_7$	$P_0$	$P_3$
$P_6$	$P_5$	$P_4$

**Figure 4.6 :**  $P_0$  Point courant ;  $P_1$ ;  $P_2$ ; .....;  $P_8$  : huit pixel voisins;  $P_1, P_3, P_5, P_7$  : quatre pixel voisin.

Un exemple de l'algorithme d'amincissement est présenté ci-dessous.

1	0	1
0	1	0
1	0	1

(A).

0	1	1
0	1	1
0	1	0

(B)

	T <sub>8</sub>	T <sub>7</sub>	T <sub>6</sub>	T <sub>5</sub>	T <sub>4</sub>	T <sub>3</sub>	T <sub>2</sub>	T <sub>1</sub>
1.T :	1	0	1	0	1	0	1	0

	T <sub>8</sub>	T <sub>7</sub>	T <sub>6</sub>	T <sub>5</sub>	T <sub>4</sub>	T <sub>3</sub>	T <sub>2</sub>	T <sub>1</sub>
	0	0	0	1	0	1	1	1

2.TM:	0	0	0	0	0	0	0	0
-------	---	---	---	---	---	---	---	---

	0	0	0	1	0	1	0	1
--	---	---	---	---	---	---	---	---

3.Q :	0	0	0	0	0	0	0	0
-------	---	---	---	---	---	---	---	---

	0	0	1	0	1	0	1	0
--	---	---	---	---	---	---	---	---

4.TR:	1	0	1	0	1	0	1	0
-------	---	---	---	---	---	---	---	---

	0	0	1	1	1	1	1	1
--	---	---	---	---	---	---	---	---

5 N<sub>1</sub>----->0 (TR) = 4

N<sub>1</sub>----->0 (TR) = 1

N<sub>1</sub> (TR) = 4

N<sub>1</sub> (TR) = 6

6. P<sub>0</sub> n'est pas enlevé

P<sub>0</sub> est enlevé.

#### 4.5 RESULTATS :

Cet algorithme d'amincissement séquentiel préserve d'une façon remarquable la forme de l'image originale, et le temps d'exécution du processus est très court.

L'inconvénient trouvé est la déconnexité en quelque régions du modèle binaire utilisé.

L'utilisation de la deuxième équation des règles de sélection, permet de produire un squelette qui est plus connexe que celui obtenu en utilisant la première équation.

1. En utilisant la première équation des règles de sélection

le nom du programme est : D.FOR (voir Annexe B).

2. En utilisant la deuxième équation des règles de sélection

le nom du programme est : D1.FOR (voir Annexe B).

## 4.6\_ AMELIORATION DE L'ALGORITHME D'AMINCISSEMENT

### SEQUENTIEL :

#### 4.6.1\_ DESCRIPTION :

L'Algorithme d'amincissement séquentiel présenté précédemment a un problème de discontinuité.

Pour minimiser cet inconvénient, on introduit dans le programme les deux première conditions données dans le nouvel algorithme d'amincissement parallèle.

$$1. \quad 2 \leq B(P_1) \leq 7$$

$$2. \quad A(P_1) = 1.$$

où  $B(P_1)$  est le nombre de voisins non nul de  $P_1$  et  $A(P_1)$  est le 01 modèle dans l'ordre  $P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9$  et  $P_2$ .

Notons que la condition (1) vérifie que  $P_1$  n'est pas un point final et la condition (2) vérifie que  $P_1$  n'est pas un point de rupture, (i.e ce n'est pas un point qui peut rompre la connexité du squelette).

#### 4.6.2\_ RESULTATS :

En introduisant les deux équations (1) et (2), le problème de discontinuité dans le squelette obtenu, utilisant l'algorithme d'amincissement séquentiel, est minimisé.

Le nom du programme est DMOD.FOR ( voir Annexe B).

**CHAPITRE 5**

**AMINCISSEMENT ET EPAISSISSEMENT  
DES MODELES BINAIRES**

## 5.1\_ NOUVEL ALGORITHME D'AMINCISSEMENT PARALLELE :

### 5.1.1\_ INTRODUCTION:

Cet algorithme a été développé pour examiner les problèmes rencontrés précédemment. Il est basé sur les opérations locales pour détecter les points du bord, les points finaux et les points de ruptures.

(i.e que le point  $P_1$  est à supprimer si c'est un point du bord, non un point final et non un point de rupture). De même cet algorithme consiste à appliquer quatre sous-itérations.

### 5.2.1\_ ALGORITHME :

Dans chaque passage, le modèle donné est balayé ligne par ligne de haut en bas et de gauche à droite. Dans chaque passage un point noir  $P_1$  est repéré si au moins une des quatres sous-itération suivantes est satisfaite :

La première sous-itération :

1.  $a_2 \leq B(P_1) \leq 7$   
b\_  $A(P_1) = 1$   
c\_  $P_2 * P_4 * P_6 = 0$   
d\_  $P_4 * P_6 * P_8 = 0$

La deuxième sous-itération :

2.  $a_2 \leq B(P_1) \leq 7$   
b\_  $A(P_1) = 1$   
c\_  $P_2 * P_6 * P_8 = 0$   
d\_  $P_4 * P_6 * P_8 = 0$

La troisième sous-itération :

3.  $a_2 \leq B(P_1) \leq 7$   
b\_  $A(P_1) = 1$   
c\_  $P_2 * P_4 * P_8 = 0$   
d\_  $P_2 * P_6 * P_8 = 0$

La quatrième sous-itération :

4.  $a_2 \leq B(P_1) \leq 7$   
b\_  $A(P_1) = 1$   
c\_  $P_2 * P_4 * P_6 = 0$   
d\_  $P_2 * P_4 * P_8 = 0$

à la fin de chaque passage tout les points repérés sont effacés.

### 5.1.3\_ EXPLICATION DES QUATRES SOUS ITERATIONS :

On commencera par l'explication de la première et la deuxième condition parcequ'elles sont les mêmes pour toutes les sous itérations, ensuite les conditions (c) et (d) pour chaque sous itération.

1- Dans la condition (a), le point final dans le squelette est préservé (c.a.d que cette condition vérifie que  $P_1$  est un point du bord (limite) mais pas un point final).

Notons que un point limite est un point noir qui a au moins un des quatres voisins blanc; et un point final est un point noir qui a au plus un des huit voisins noir.

2- La condition (b) vérifie que le point  $P_1$ , qui est à effacer, n'est pas un point de rupture (i.e ça suppression ne doit pas rompre la connexité du modèle originale).

3- Les conditions (c) et (d) pour la première sous-itération :

$$P_2 * P_4 * P_6 = 0$$

$$P_4 * P_6 * P_8 = 0$$

Dans la condition (3) le point  $P_1$  à effacer peut être un point limite "est" ou un point limite "sud" ou un point limite "nord-est".

Les solutions de cet ensemble d'équations sont :

$$P_4 = 0$$

$$\text{ou } P_6 = 0$$

$$\text{ou } P_2 = 0 \text{ et } P_8 = 0$$

4- Les conditions (c) et (d) pour la deuxième sous itération :

$$P_2 * P_6 * P_8 = 0$$

$$P_4 * P_6 * P_8 = 0$$

Dans la condition (4) le point  $P_1$  à enlever peut être un point limite "sud" ou un point limite "ouest" ou un point limite "nord-est".

Les solutions de cet ensemble d'équations sont :

$$P_6 = 0$$

$$\text{ou } P_8 = 0$$

$$\text{ou } P_2 = 0 \text{ et } P_4 = 0$$

5- Les conditions (c) et (d) pour la troisième sous itération :

$$P_2 * P_4 * P_8 = 0$$

$$P_2 * P_6 * P_8 = 0$$

Dans la condition (5) le point  $P_1$  à enlever peut être un point limite "nord" ou un point limite "ouest" ou un point limite "sud-est".

Les solutions de cet ensemble d'équation sont :

$$P_2 = 0$$

$$\text{ou } P_8 = 0$$

$$\text{ou } P_4 = 0 \text{ et } P_6 = 0$$

6- Les conditions (c) et (d) pour la quatrième sous itération :

$$P_2 * P_4 * P_6 = 0$$

$$P_2 * P_4 * P_8 = 0$$

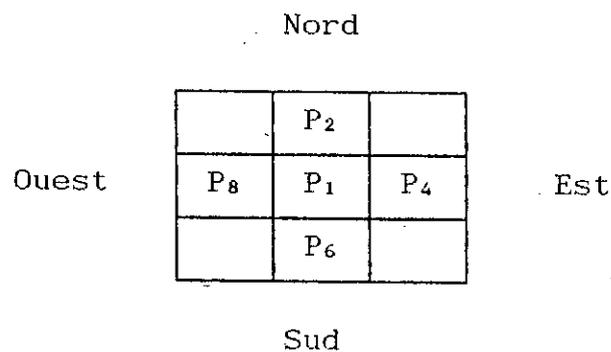
Dans la condition (6) le point  $P_1$  à enlever peut être un point limite "nord" ou un point limite "est" ou un point limite "sud-ouest".

Les solutions de cet ensemble d'équations sont :

$$P_2 = 0$$

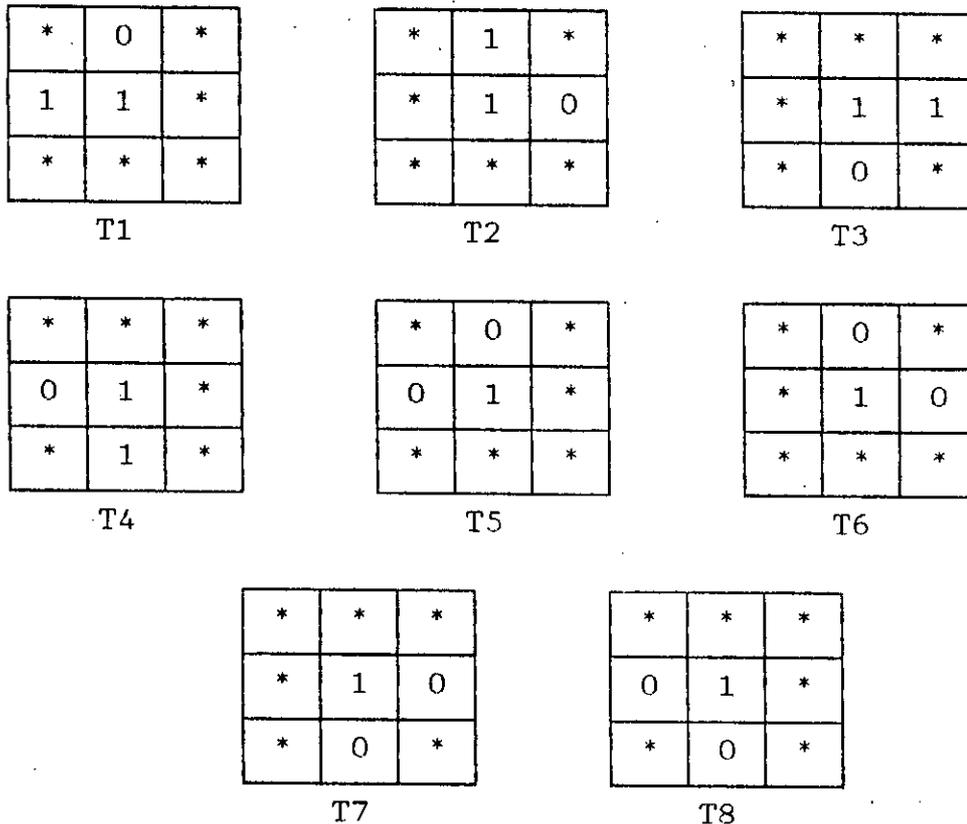
$$\text{ou } P_4 = 0$$

$$\text{ou } P_6 = 0 \text{ et } P_8 = 0$$



**Figure 5.1 :** Localisation des points considérés.

l'Algorithme décrit au dessus peut être aussi développé en utilisant une méthode itérative comme il est montré ci dessous. Dans les conditions (c) et (d), et pour chaque sous-itération, les huit fenêtres suivantes peuvent être créés comme il est illustré à la fig (5.2).



**Figure 5.2 :** Ensemble de fenêtres utilisé dans le processus : la valeur 1 correspond au pixel objet, la valeur 0 correspond à l'arrière plan et \* peut être 1 ou 0.

Si les deux conditions :

1.  $2 \leq B(P_1) \leq 7$
2.  $A(P_1) = 1$

Sont satisfaites et au moins une des huit fenêtres  $T_1, T_2, T_3, T_4, T_5, T_6, T_7$  et  $T_8$  données ci dessus correspond à celle donnée dans l'image alors le pixel central  $P_1$  est repéré.

Dans chaque passage le point  $P_1$  est repéré si au moins l'une des sous itérations est satisfaite (première procédure) ou la ressemblance est obtenue (deuxième procédure). à la fin de chaque passage tout les points repérés sont effacés. Le processus est répété jusqu'a ce qu'il n'y est aucun pixel à effacer du modèle, ainsi le squelette final est obtenu.

l'Organigramme du nouvel algorithme d'amincissement parallèle est illustré à la figure (5.3).

#### **5.1.4\_ RESULTATS :**

En utilisant l'une ou l'autre des deux procédures (la première procédure ou la deuxième procédure); cet algorithme donne d'excellent résultats en préservant la forme de l'image originale, la connexité et un contour insensible au bruit (pas de cerifs).

Le nom du programme est : ALGSUB4.FOR (voir Annexe B).

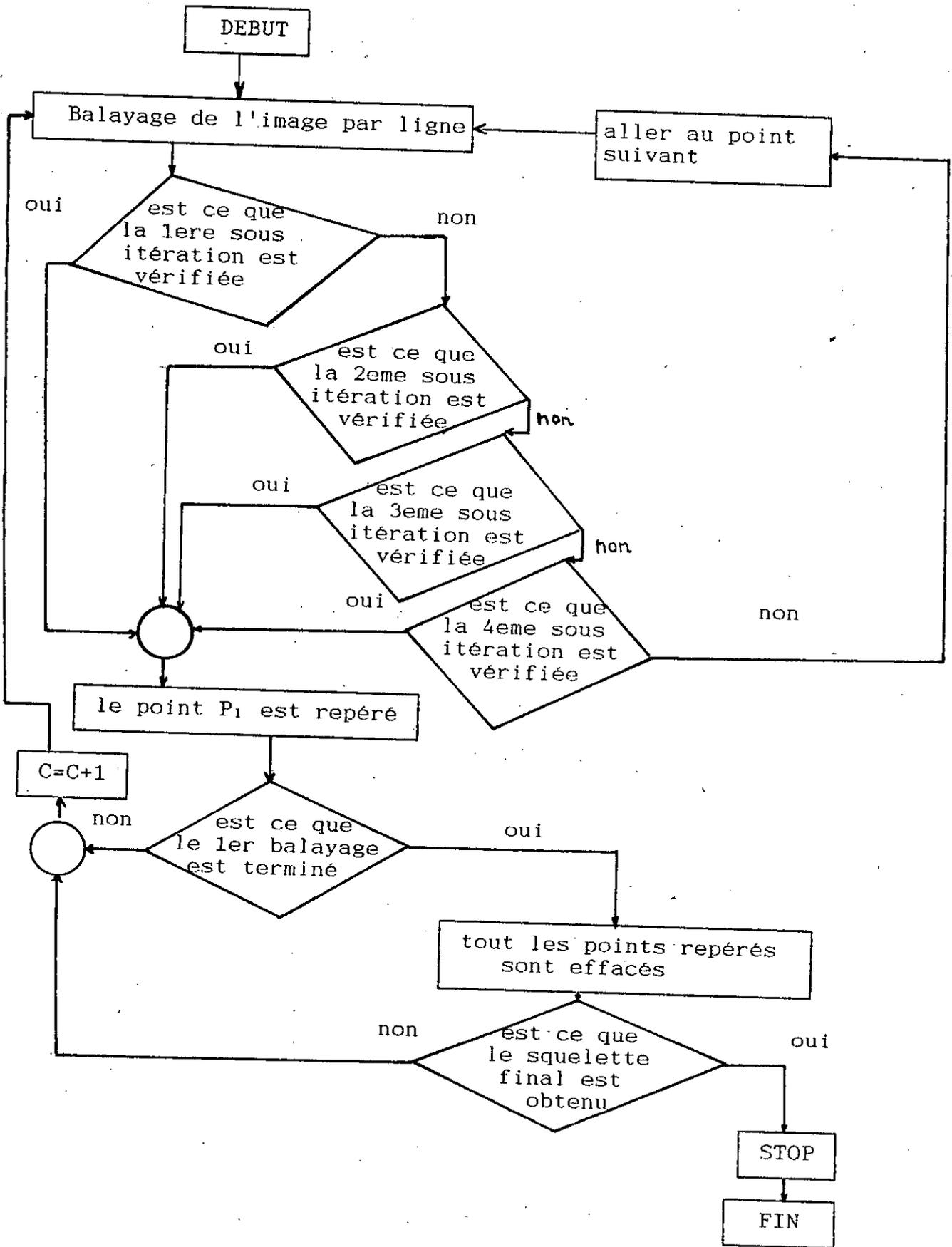


Figure 5.3 : Organigramme du nouvel algorithme d'amincissement parallèle.

## 5.2\_ ALGORITHME DE COMPARAISON MODIFIE :

### 5.2.1\_ INTRODUCTION :

Dans un processus de transformation d'image, il n'est pas seulement important de développer de nouvelles techniques mais aussi d'améliorer les techniques qui existent déjà.

Le développement des deux algorithmes parallèle et séquentiel pour chaque techniques est nécessaire, l'image procédée peut être traitée donc par un calcul séquentiel ou un autre parallèle.

### 5.2.2\_ ALGORITHME :

L'Algorithme de comparaison modifié est un autre algorithme d'amincissement parallèle dans lequel un ensemble de huit fenêtres notées par  $A_1$ ,  $A_2$ ,  $A_3$ ,  $A_4$ ,  $B_1$ ,  $B_2$ ,  $B_3$  et  $B_4$  et deux images (l'image courante pour laquelle les fenêtres sont comparés et l'image traitée, de même taille, présente une fois que les fenêtres seront fermées) sont utilisés dans le processus.

Les pixels sont enlevés par comparaison de chaque pixel de valeur 1 et de ses voisins dans l'image courante avec l'ensemble des fenêtres. Dans la fenêtre les zéros doivent coïncidés avec les pixels de valeur égal à 0, les uns doivent coïncidés avec les pixels de valeur égale à 1, et l'astérisque peut coïncidé avec un pixel de valeur égale à 1 ou 0 dans l'image courante. Initialement l'image courante et l'image traitée sont des copies identiques de l'image originale donnée. Au début, la fenêtre  $A_1$  est comparé à tous les pixels de valeur 1 et leurs voisins dans l'image courante.

Si la ressemblance est obtenu, le pixel central correspondant de l'image traitée est effacé (changé en un pixel de valeur égale à 0). Après exécution du processus avec la fenêtre  $A_1$ , l'image courante est écartée et l'image traitée devient la nouvelle image courante. Le processus est répété avec les fenêtre  $B_1$ ,  $A_2$ ,  $B_2$ ,  $A_3$ ,  $B_3$ ,  $A_4$ , et  $B_4$  dans cet ordre en formant un cycle complet. lorsque aucun pixel n'est effacé durant le processus complet du cycle, la procédure s'arrête. La figure (5.4) montre l'organigramme de l'algorithme de comparaison modifié utilisé.

### 5.2.3\_ RESULTATS :

La méthode de comparaison utilisée en un processus parallèle donne les résultats suivants :

- La forme de l'image originale est très bien préservée.
- Le squelette produit a une bonne connexité.
- l'Algorithme ne produit que peu de cerifs sur le modèle traité.

Le nom du programme est : TEMP2.FOR (Voir Annexe B).

### 5.2.3\_ RESULTATS :

La méthode de comparaison utilisée en un processus parallèle donne les résultats suivants :

- La forme de l'image originale est très bien préservée.
- Le squelette produit a une bonne connexité.
- l'Algorithme ne produit que peu de cerifs sur le modèle traité.

Le nom du programme est : TEMP2.FOR (Voir Annexe B).

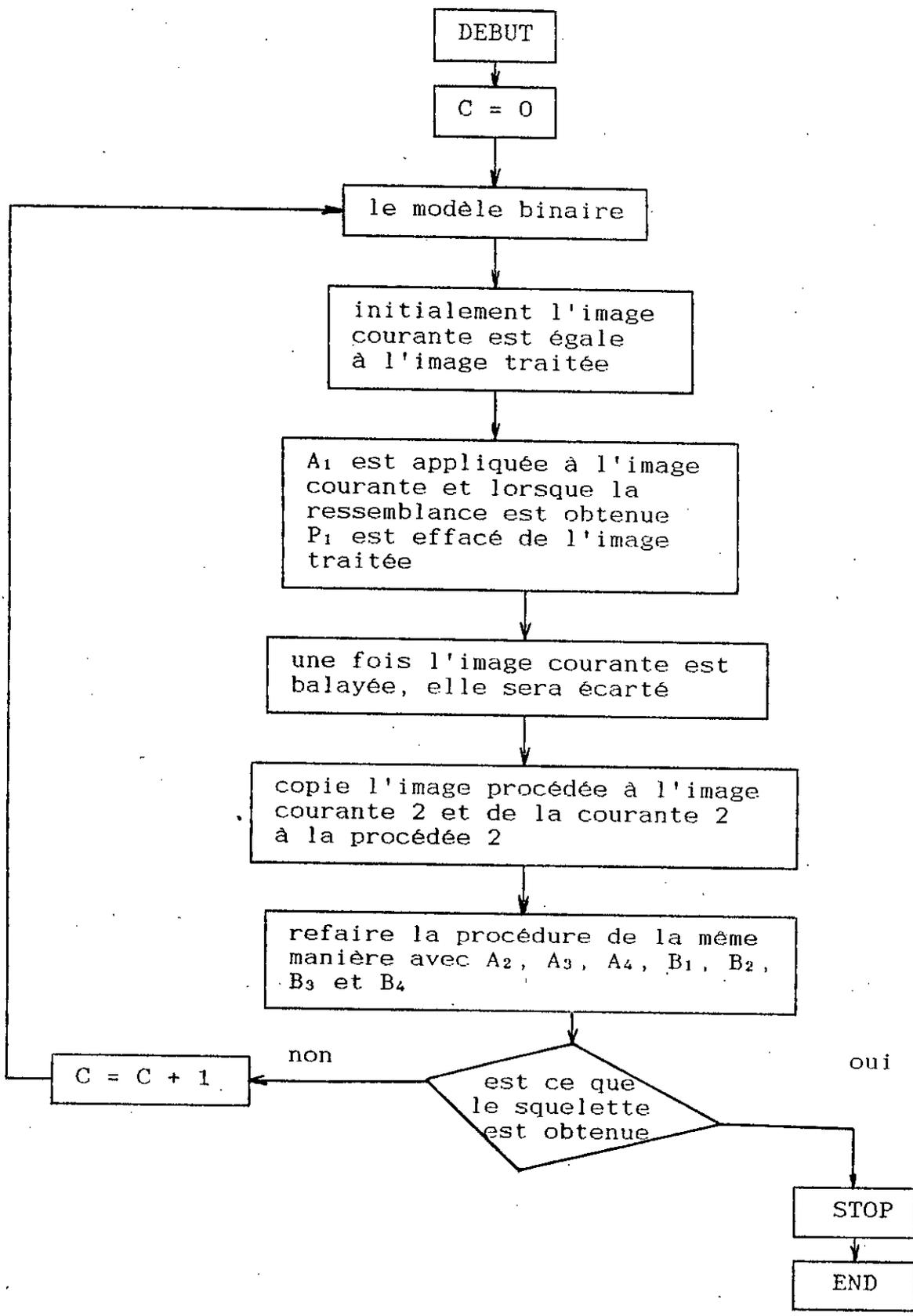


Figure 5.4 : Organigramme de l'algorithme de comparaison modifié.

## **5.3- COMPARAISON GENERALE :**

### **5.3.1\_ L'ALGORITHME DE HILDITCH ET LE NOUVEL**

#### **ALGORITHME D'AMINCISSEMENT PARALLELE :**

La version simplifiée de HILDITCH ne peut pas préserver la forme du modèle original.

En utilisant le nouvel algorithme d'amincissement parallèle qui est basé sur les quatre sous itération on obtient d'excellents résultats en préservant la connexité, la forme du modèle original et en supprimant les bruits au contour.

### **5.3.2\_ L'ALGORITHME DE COMPARAISON ET SON AMELIORATION:**

En utilisant les huit fenêtres (masques) données précédemment et seulement une image dans le processus, on obtient un squelette mais pas au milieu et avec des branches sur le modèle considéré.

En utilisant les mêmes fenêtres et deux images (l'image précédente et l'image courante); on obtient d'excellents résultats en préservant la forme de l'image originale, la connexité et le bruit au contour.

### **5.3.3\_ ALGORITHME D'AMINCISSEMENT SEQUENTIEL ET SON**

#### **AMELIORATION :**

L'algorithme séquentiel qui est basé sur la DT, les règles de sélection, l'algorithme de liaison et l'algorithme d'amincissement, est le meilleur algorithme concernant le temps d'exécution du processus, préserve bien la forme de l'image originale mais a le problème de la discontinuité. En ajoutant les deux conditions comme il a été expliqué précédemment, le problème de discontinuité est minimisé.

Une comparaison générale entre les différents algorithmes est donnée dans le tableau suivant :

ALGORITHMES	AVANTAGES	INCONVENIENT
ALGORITHME DE Comparaison	Bonne connexité	Présence de cerifs
ALGORITHME DE NACCACHE	La forme est bien préservée	Discontinuité et cerifs
VERSION Simplifiée de HILDITCH	Bonne connexité	Les propriétés de la forme ne sont pas préservés
ALGORITHME D'AMIN-CISSEMENT séquentiel	Excellent concernant le temps du processus	Discontinuité
ALGORITHME DE Comparaison modifiée	excellent résultats forme préservée, la connexité conservée et bruit au contour enlevé	Le temps du processus est long
NOUVEL ALGORITHME d'amincissement parallèle	même avantage que l'algorithme de comparaison modifiée	Pas d'inconvénients

## **5.4 \_ ALGORITHME D'ÉPAISSISSEMENT :**

### **5.4.1 \_ INTRODUCTION :**

Le processus d'épaississement est l'inverse du processus d'amincissement, il permet la reconstruction du modèle (objet sur l'image) initial à partir du squelette.

L'Algorithme d'épaississement est essentiellement appliqué sur des modèles squelettes ayant été le produit d'algorithmes métriques.

Dans ce qui suit on présentera cet algorithme d'épaississement.

### **5.4.2 \_ ALGORITHME :**

L'Algorithme consiste essentiellement à exécuter itérativement des balayages sur le modèle considéré; à chaque passe, les points de l'arrière-plan qui sont liés aux extrémités du modèle sont changés en points objets.

Pour un premier balayage l'image est parcourue ligne par ligne, de gauche à droite et de haut en bas. Pour toute variation de 0-1 (i.e de l'arrière-plan au point objet) ou bien de 1 à 0 (i.e du pixel objet à un pixel de l'arrière-plan) on étiquette l'arrière-plan avec la valeur(2) ce qui veut dire que la valeur (2) va appartenir à l'objet après le deuxième balayage.

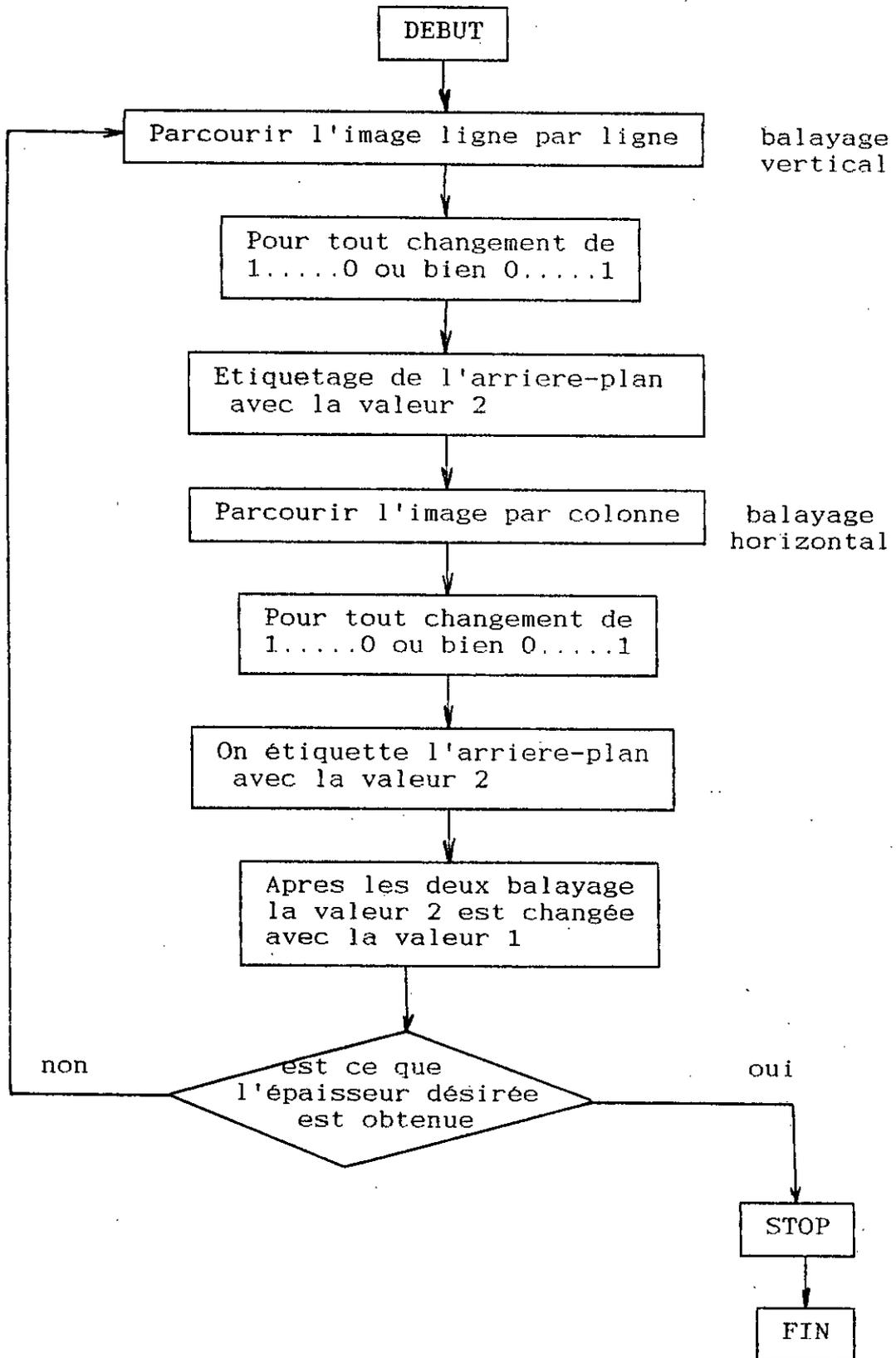
Pour le deuxième balayage, on parcourt l'image par colonne de haut en bas et de gauche à droite; pour chaque changement de 0-1 ou bien de 1-0 on étiquette l'arrière-plan par la valeur (2). Une fois le deuxième balayage terminé tous les pixels ayant l'étiquette (2) sont alors changés à la valeur 1 qui est la valeur des points objets.

Le processus est répété de manière itérative jusqu'à obtention de l'épaisseur désirée.

L'Organigramme de l'algorithme d'épaississement est illustré à la figure (5.5).

### **5.4.3 \_ RESULTATS :**

En utilisant cette méthode pour l'épaississement on obtient d'excellent résultats et de belles formes avec l'épaisseur choisi. Le nom du programme est : THICK.FOR (Voir Annexe B).



**Figure 5.5 :** Organigramme de l'algorithme d'épaississement .

## **CHAPITRE 6**

# **PRESENTATION DES RESULTATS**

## RESULTATS EXPERIMENTAUX

Les programmes que nous avons préparé dans cette étude (Voir Annexe B) sont basés essentiellement sur les trois différentes méthodes suivantes :

- 1- La méthode de comparaison.
- 2- La méthode des opérations locales.
- 3- La méthode de la "DT".

Ces programme utilisent deux processus différents, le processus parallèle et le processus séquentiel, ainsi que deux équations mathématiques différentes utilisées particulièrement pour les méthodes métriques.

Les résultats de ces différentes méthodes sont illustrés par quelque exemples selon des forme particulières dans les figures suivantes :

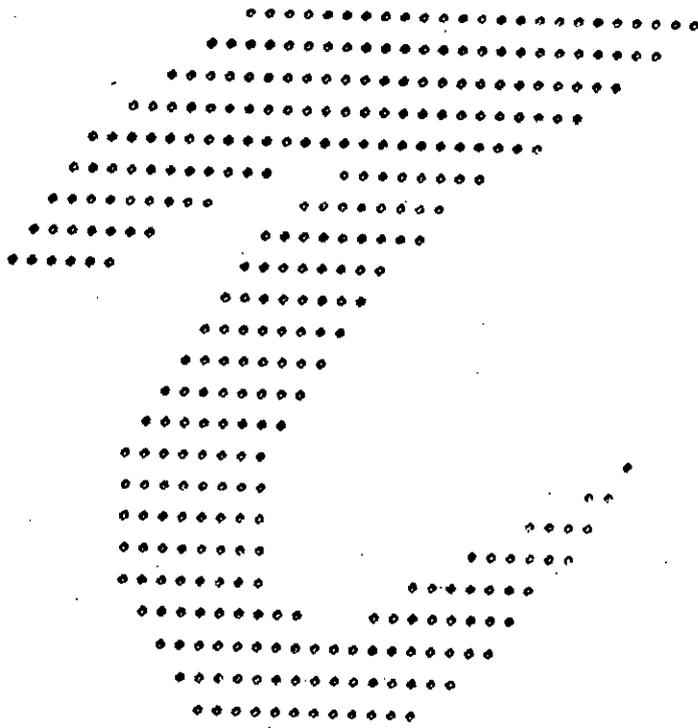


Figure 6.1 : La forme originale, (caractère To) .

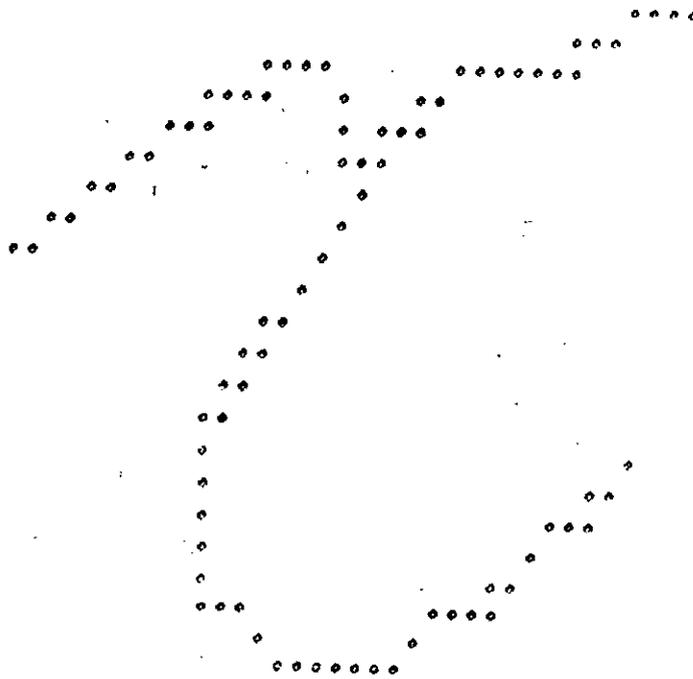


Figure 6.2 : Squelette obtenu par la méthode de comparaison (processus séquentiel).

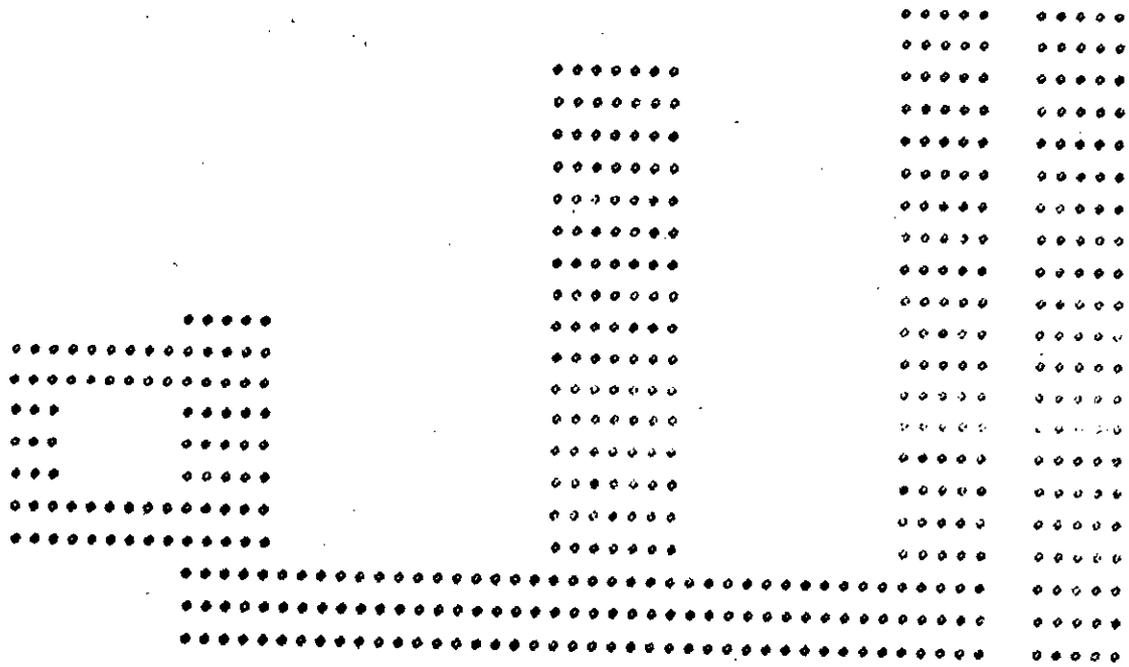


Figure 6.3 : La forme originale;

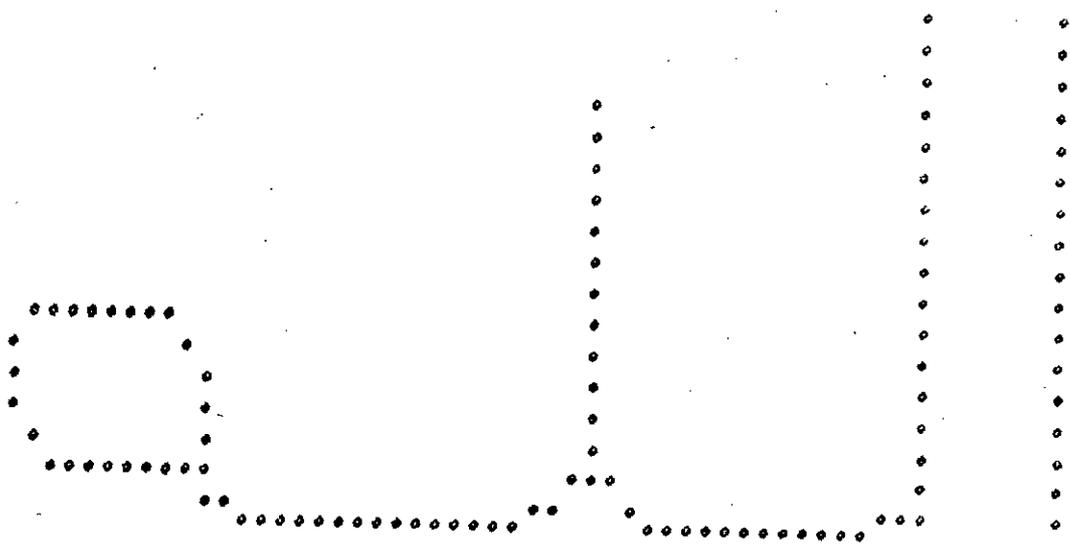


Figure 6.4 : Squelette obtenu par la méthode de comparaison (processus parallèle).

Figure 6.6 : Squelette obtenu par la méthode de la "DT" (équation "a").

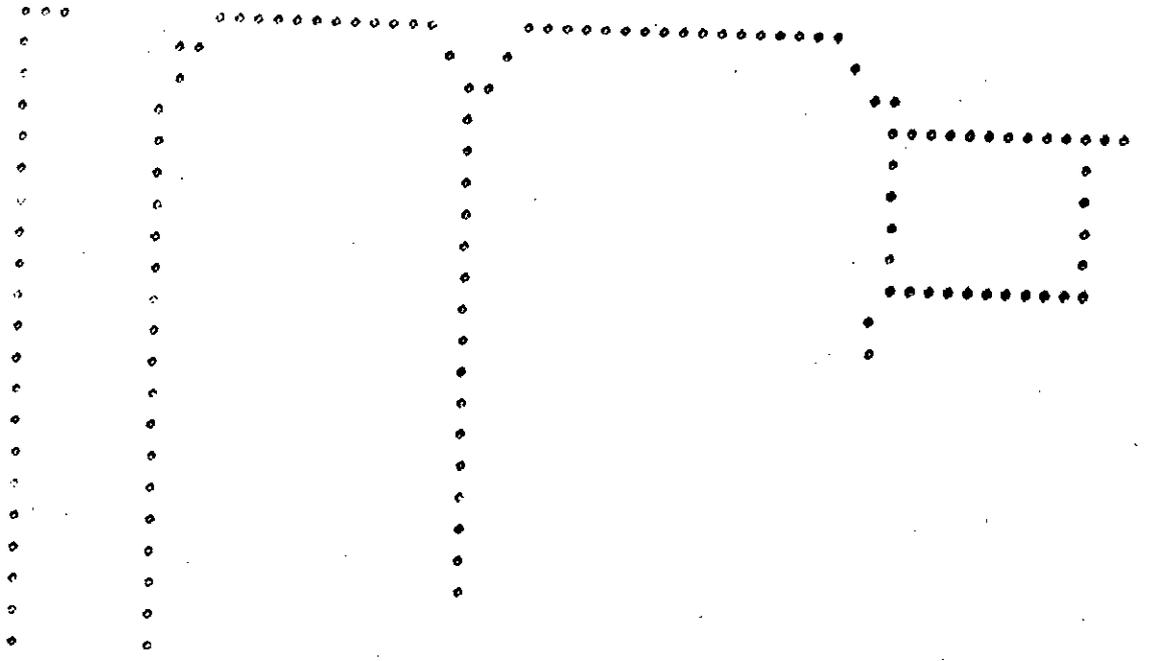
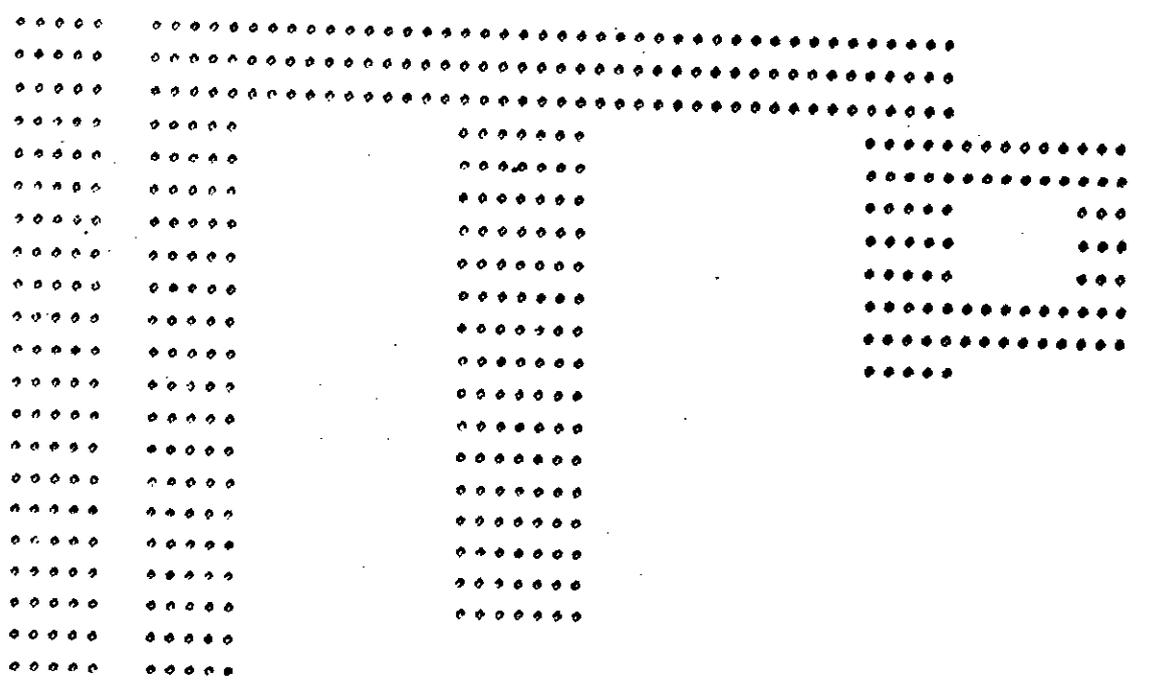


Figure 6.5 : La forme originale;



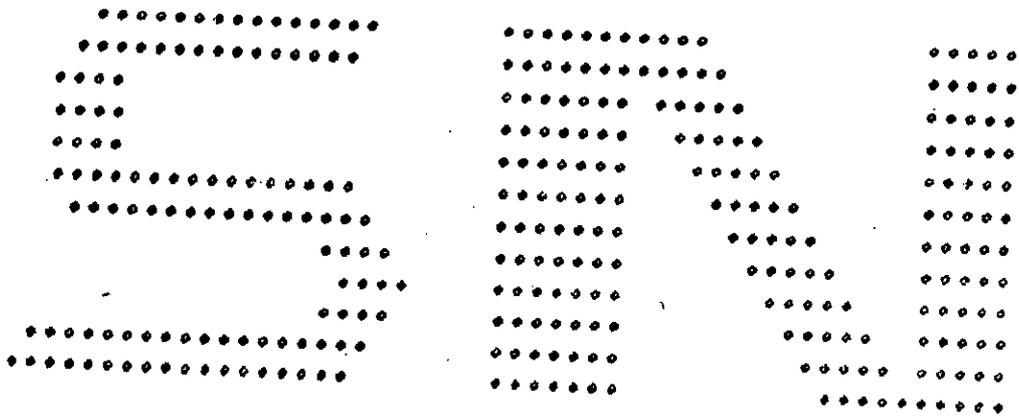


Figure 6.7 : La forme originale;

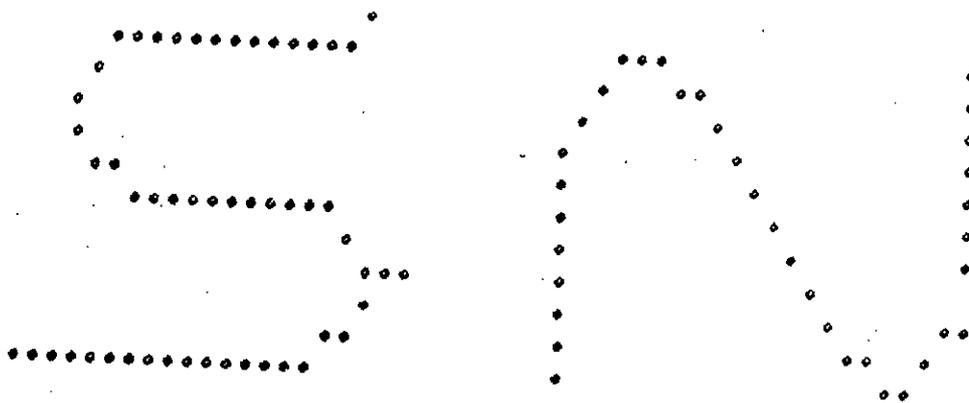


Figure 6.8 : Squelette obtenu par la méthode des opérations locales (processus parallèle).

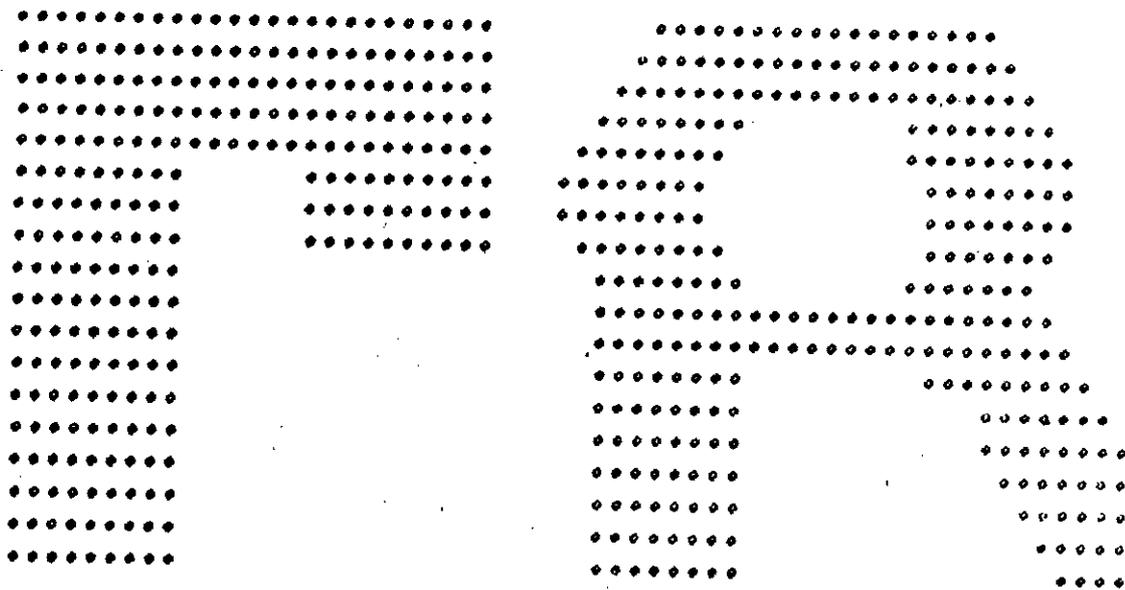


Figure 6.9 : La forme originale;

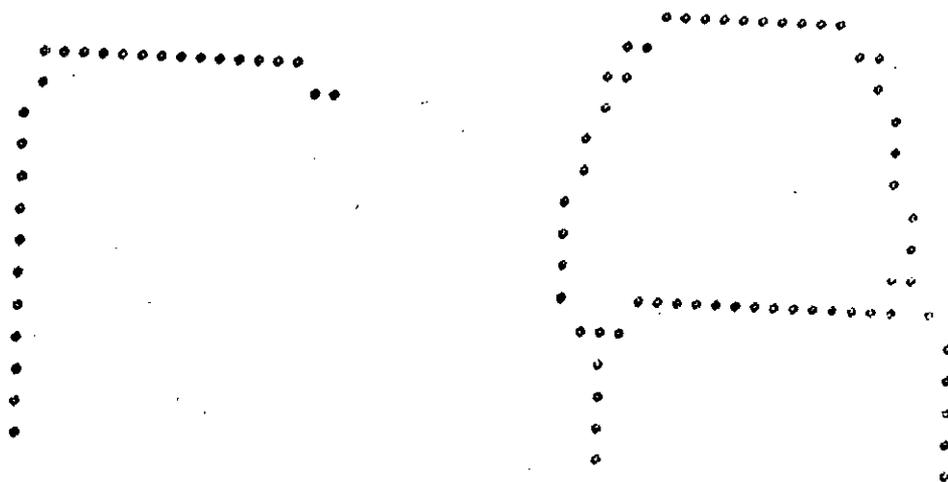


Figure 6.10 : Squelette obtenu par la méthode des opération locales (processus séquentiel).

## **CONCLUSION GENERALE**

## CONCLUSION GENERALE

Dans cette étude, nous avons présenté multiple algorithmes d'amincissement, nous avons discuté toutes les conditions nécessaire que doit vérifier un point noir avant d'être éliminé du modèle considéré.

L'Algorithme d'amincissement parallèle préserve de façon remarquable la forme de l'image originale, mais produit deux problèmes :

1- Présence de cerif sur le squelette .

2- Le squelette final n'est pas toujours connexe.

Pour la méthode de comparaison qui utilise l'ensemble de huit fenêtres, nous avons montré que si une image seulement est utilisée dans le processus, on obtient un squelette mais pas au centre, et en utilisant deux images (l'image courante et l'image traitée) on obtient d'excellent résultats en préservant la forme de l'image originale, la connexité et le bruit au contour.

Nous avons montré également que la version simplifiée de l'algorithme d'amincissement permet de supprimer après la dernière itération, tous les points qui ont été repérés après la vérification des quatres conditions.

Par l'utilisation du nouvel algorithme d'amincissement parallèle qui consiste à vérifier une des quatres sous-itérations (chacune est divisée en quatres conditions), on obtient d'excellent résultats en préservant la forme de l'image originale, la connexité et le bruit au contour.

La méthode de la transformation de la distance qui est la meilleurs concernant le temps d'exécution et qui préserve bien la forme de l'image originale, a un problème de discontinuité qui est éliminé en utilisant les deux conditions comme il a été expliqué précédemment .

A la fin de cette étude, nous avons présenté un algorithme d'épaississement qui donne de très bon résultats.

## **REFERENCES BIBLIOGRAPHIQUES**

## BIBLIOGRAPHIE

- [01] JEAN-JACQUES TOUMAZET : "traitement de l'image sur micro-ordinateur", SYBEX, 1987.
- [02] ANDRE MARION : "Introduction aux techniques de traitement d'image", EYROLLES, 1987.
- [03] C. RAFAEL GONZALES - PAUL WINTZ : "Digital image processing", KOREAN Student Edition.
- [04] ALAIN FAURE : "Perception et reconnaissance des formes", Edition Tests.
- [05] J.C. SIMON : "La reconnaissance des formes par algorithmes", Edition Masson.
- [06] M. COSTER et J.L. CHERMANT : "Precis d'analyse d'image".
- [07] AFCET. INRA : "La reconnaissance des formes et intelligence artificielle", Tome 1, 6<sup>ème</sup> Congres exposition DUNOD.
- [08] COGNET : "Visualisation et traitement d'images".
- [09] N. BENAÏFA : "Etude de la methode structurelle pour la reconnaissance des formes appliquée aux caractères arabes manuscrits" ENP, Alger 1993.
- [10] C. ARCELLI et G. SANNIT DI BAJA : "A Width independent fast thinning algorithm" IEEE trans. on pattern recognition and machine intelligence, Vol 7, N°=4, July 1985.
- [11] PELEG and A. ROSEN-FELD, "A MIN-MAX Medial axis transformation", IEEE trans on pattern anal. mach in intell. 3, pp. 208-210, 1981.
- [12] E.R. DAVIES and A.P.N. plummer, "thinning algorithms : a critique and new methodology", pattern recognition. 14, pp. 53-63, 1981.
- [13] R.M. SMITH, " Computer processing of line images : A survey", pattern recongnition, vol. 20 n°. 1, pp. 7-15, 1987.
- [14] T. WAKAYAMA, "A core line tracing algorithm based on maximal square moving", IEEE Transactions on pattern recognition and machine intelligence, vol pami. 3, pp. 134-143, 1981.

- [15] R.STEFANELLI and A.ROSENFELD,"some parallel thinning algorithms for digital picture", J.ACM18, pp.255-264, April 1971.
- [16] A.ROSENFELD and L.S. DAVIS, " A note on thinning", IEEE transaction on systems, man and cybernetics, pp.236-238, March 1976.
- [17] H.TAMURA, "A comparaison of line thinning algorithms from digital geometry viewpoint", proc.4th. Jt conf on pattern recognition, Kyoto, Japan, pp.715-719, 1978.
- [18] C.C.LEE,Modified distance transform and linking algorithm for image skeletonization", spie proceeding, vol.415, coherent infrared radar systems and application(II), arlington, pp.4-8, April 1983.
- [19] C.C.LEE, "A sequential thinning algorithm for image skeletonization", spie proceeding, vol.415, coherent infrared radar systems and application (II), arlington, 1984.
- [20] J.T. KUEHN, J.A FESSLER, and H.J. SIEGEL, "parallel image thinning and vectorization on pasm", proc.IEEE, pp.368-374, 1985.
- [21] C.ARCELLI and G.S DI-BAJA, " a thinnig algoritm based on pomise", pattern recognition society, pp.225-235, non 1980.
- [22] C.ARCELLI, L.CORDELLA and S.LEVIALDI "parrallel thinning of binary pictures", electronics letters, vol.11, pp.138-149, July 1976.
- [23] C.J. HILDITCH, "comparaison of thinning algorithms on a parallel processus", Image and vision processing, vol.1, pp. 115-132, Aug 1983.
- [24] N.J. NACCACHE and R.SHINGHAL," An investigation into skeletonization approach of HILDITCH", pattern recognition, vol.17 N°.3, pp.279-284, 1984.
- [25] A.R.DILL, M.D.LEVINE, and P.B.NOBLE, " Multiple resolution skeletons", IEEE transaction on pattern recognition and machine intelligence, vol PAMI.9 N°.4, pp.495-503, July 1987.
- [26] SEYMOUR LIPS-CHUTZ-ARTHUR POE : "Programmation Fortran", theorie et applications.
- [27] JEAN-CLAUDE ARMICI : "Programmer en turbo pascal", Edition LI.
- [28] J.RIVIERE : " Turbo pascal et son environnement", DUNOD informatique.

[29] M.DUBESSET-J.VIGNES : " Fortran le langage normalisé", Edition Technip 1991.

[30] SERGE CASTAN : "Vision par ordinateur".

[31] BERNARD PEROCHE - JACQUELINE ARGENCE - GHAZAN-FAR-POUR - MICHELUCCI : " La synthese d'images", traité des nouvelles technologies - Edition HERMES.

[32] T.PAVILIDIS, " A flexible parallel thinning algorithm", proc. IEEE comput. SOC. CONF. on pattern recognition and image processing, pp.162-167, Aug 1981.

[33] Y.N. CHU and C.Y. SVEN, " An alternate smoothing and stripping algorithm for thinning digital binary pattern", signal processing, vol.N°3, pp.207-222, 1986.

[34] N.J. NACCACHE and R.SHINGHAL, "SPTA : A proposed algorithm for thinning binary pattern", IEEE transcation on systems, Man, and cybernetics, vol. SMC.14 N°3, pp. 409-418, 1984.

**ANNEXE A**

**L'HISTOGRAMME**

## L' HISTOGRAMME

### DEFINITION :

l'Histogramme des niveaux de gris d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris dans l'image.

En abscisse sont représentés les niveaux de gris.

En ordonnée est affiché le nombre de pixels affecté par chaque niveau de gris. Notons que l'on peut aussi exprimer l'ordonnée en pourcentage : niveau de gris affecté/nombre total de pixels de l'image.

l'Histogramme permet de donner un grand nombre d'informations sur la distribution des niveaux de gris de l'image, ce qui est plus communément appelé la dynamique de l'image. Il permet notamment d'analyser entre quelles bornes est répartie la majorité des niveaux de gris dans le cas d'une image trop claire ou trop foncée[1].

**ANNEXE B**

**PRESENTATION DES PROCEDURES  
FORTRAN RELATIVES AUX  
DIFFERENTES METHODES**

## L'ALGORITHME DE COMPARAISON ET SON AMELIORATION :

### PROCEDURES FORTRAN RELATIVES A CETTE METHODE :

#### PROCEDURE TEMP1.FOR (ALGORITHME DE COMPARAISON)

Cette procédure permet d'utiliser l'algorithme d'amincissement séquentiel, une seule image et un ensemble de huit(08) fenêtres sont utilisées dans le processus.

#### PROCEDURE TEMP2.FOR (ALGORITHME DE COMPARAISON AMELIORE)

Cette procédure permet d'utiliser l'algorithme d'amincissement parallèle; deux images sont utilisées dans le processus, "l'image courante " sur laquelle l'ensemble des 8 fenêtres sont comparées et "l'image traitée" une fois les fenêtres ont été comparés.

## L'ALGORITHME DES OPERATIONS LOCALES :

### PROCEDURES FORTRAN RELATIVES A CETTE METHODE:

#### PROCEDURE ALH.FOR (LES OPERATIONS LOCALES PARALLELES)

Cette procédure utilise le processus d'amincissement parallèle, un pixel  $P_1$  est repéré si 6 tests sont vérifiés. à la fin du balayage tout les points repérés sont effacés.

#### PROCEDURE ALGSUB10.FOR (LES OPERATIONS LOCALES SEQUENTIELLES)

Cette procédure utilise le processus d'amincissement séquentiel où quatre (04) conditions sont à satisfaire pour qu'un point image  $P_1$  soit effacé.

a).  $2 \leq B(P_1) \leq 7$ .

b).  $A(P_1) = 1$

c).  $P_2 * P_4 * P_6 = 0$

d).  $P_2 * P_4 * P_8 = 0$

**PROCEDURE ALGSUB1.FOR (LES OPERATIONS LOCALES  
PARALLELES)**

Cette procédure utilise le processus d'amincissement parallèle, un point  $P_1$  est repéré par une autre valeur, à la fin du balayage tous les points repérés sont effacés.

**PROCEDURE ALGSUB4.FOR (ALGORITHME DES OPERATIONS  
LOCALES AMELIORE)**

Cette procédure utilise les opérations locales pour détecter les points limites, les points finaux et les points de rupture. Cet algorithme est divisé en quatre(04) sous itérations, chaque sous itération se compose de quatre(04) conditions.

1<sup>ere</sup> Sous itération :

- a).  $2 \leq B(P_1) \leq 7.$
- b).  $A(P_1) = 1$
- c).  $P_2 * P_4 * P_6 = 0$
- d).  $P_4 * P_6 * P_8 = 0$

2<sup>eme</sup> Sous itération :

- a).  $2 \leq B(P_1) \leq 7.$
- b).  $A(P_1) = 1$
- c).  $P_2 * P_6 * P_8 = 0$
- d).  $P_4 * P_6 * P_8 = 0$

3<sup>eme</sup> Sous itération :

- a).  $2 \leq B(P_1) \leq 7.$
- b).  $A(P_1) = 1$
- c).  $P_2 * P_4 * P_8 = 0$
- d).  $P_2 * P_6 * P_8 = 0$

4<sup>eme</sup> Sous itération :

- a).  $2 \leq B(P_1) \leq 7.$
- b).  $A(P_1) = 1$
- c).  $P_2 * P_4 * P_6 = 0$
- d).  $P_2 * P_4 * P_8 = 0$

## L'ALGORITHME DE LA "DT" :

### PROCEDURES FORTRAN RELATIVES A CETTE METHODE :

#### PROCEDURE D.FOR (ALGORITHME D'AMINCISSEMENT SEQUENTIEL)

Cette procédure utilise la transformation de la distance, elle s'effectue en une succession d'opérations.

- a). Application séquentielle de la "DT".
- b). Application de la règle de sélection utilisant l'équation (A).
- c). Application de l'algorithme de liaison.
- d). Application de l'algorithme d'amincissement.

#### PROCEDURE D1.FOR (ALGORITHME D'AMINCISSEMENT SEQUENTIEL)

Cette procédure utilise la transformation de la distance, elle s'effectue en une succession d'opérations.

- a). Application séquentielle de la "DT".
- b). Application de la règle de sélection utilisant l'équation (B).
- c). Application de l'algorithme de liaison.
- d). Application de l'algorithme d'amincissement.

#### PROCEDURE DMOD.FOR (ALGORITHME D'AMINCISSEMENT AMELIORE)

Les deux procédures précédentes D.FOR et D1.FOR produisent des discontinuités sur les modèles binaires traités. Pour minimiser ce problème, deux conditions ont été ajoutés dans cette procédure.

- a).  $2 < B(P_1) < 7$ .
- b).  $A(P_1) = 1$

## L'ALGORITHME D'EPAISSISSEMENT :

### PROCEDURE FORTRAN RELATIVE A CETTE METHODE:

#### PROCEDURE THICK.FOR :

Cette procédure permet la reconstruction d'une image binaire à partir du squelette, elle utilise le processus séquentiel.