

BEUCHER  
(avec corrections)

CONFIDENTIEL

N-962

MORPHOLOGIE MATHÉMATIQUE SUR  
IMAGES A NIVEAUX DE GRIS ET  
PROCESSEURS BINAIRES

-----

Serge BEUCHER

FONTAINEBLEAU

MAI 1985

## MORPHOLOGIE MATHÉMATIQUE SUR IMAGES A NIVEAUX DE GRIS ET PROCESSEURS BINAIRES

L'utilisation de transformations morphologiques sur des images à niveaux de gris ne peut s'envisager que par le biais d'une des trois possibilités suivantes :

- L'usage d'un analyseur d'images possédant des processeurs d'images de gris câblés (VICOM, VISIOMAT, etc...)
- L'usage d'un logiciel de morphologie mathématique pour images de gris implanté sur un ordinateur classique (MORPHOLOG, etc...).
- L'usage combiné d'un analyseur d'images binaires avec processeurs câblés binaires et d'un logiciel adapté de traitement numérique (NS 1500, Leitz-TAS, etc...).

Cette dernière solution présente un certain nombre d'avantages. Par rapport à la première solution, et compte tenu du prix actuel des matériels, le coût d'une telle opération est relativement réduit (mais cependant il dépend beaucoup de l'analyseur binaire utilisé). Par rapport à la deuxième solution, cette façon de faire permet d'espérer des vitesses de traitement similaires à celle obtenues sur des ordinateurs performants. De plus, les opérations binaires étant effectuées par les processeurs câblés et non plus par un logiciel spécialisé, l'aspect pénalisant constitué par les vitesses de traitement de ce logiciel pour les opérations binaires est supprimé.

Cependant l'implantation de cette configuration n'est pas sans poser des problèmes à la fois logiciels (architecture du programme) et matériels (visualisation souvent pas adaptée au traitement numérique, taille de la mémoire images, etc...). Il convient donc de passer en revue ces divers problèmes afin de dégager les meilleures solutions possibles.

Cette note a pour objet de passer en revue les différents algorithmes de traitement numérique et leur implantation sur un analyseur d'images binaires type NS 1500. Les problèmes matériels liés à cette implantation seront également abordés.

#### LES TRANSFORMATIONS MORPHOLOGIQUES SUR IMAGES DE GRIS

Une librairie de transformations morphologiques numériques doit contenir un certain nombre de transformations de base sans lesquelles il serait vain d'espérer réaliser des opérations intéressantes. Une liste non exhaustive de ces transformations comprendra l'érosion et la dilatation numérique, l'épaississement et l'amincissement, les opérations arithmétiques et booléennes (addition, soustraction, sup, inf, etc...), divers utilitaires (seuillage par exemple) etc...

La réalisation pratique de ces transformations peut être envisagée de différentes manières. Cependant, l'architecture de la machine utilisée impose de telles contraintes que la solution optimale dépend rarement de l'utilisateur. Pour s'en convaincre, envisageons divers protocoles d'implantation :

- On pourrait utiliser directement les valeurs de gris de l'image et effectuer les opérations entre ces valeurs : l'érosion consisterait alors à calculer simplement l'inf entre les sept valeurs de gris du voisinage hexagonal de taille 1 de chaque point. Cette façon de faire implique que les images soient représentées en mémoire sous forme d'un tableau de valeurs de gris. L'organisation des mémoires-images d'un processeur binaire ne correspond bien évidemment pas à ce schéma.
- Une autre solution consisterait à effectuer les opérations seuil à seuil. On sait que tout seuil de l'érodé numérique n'est autre que l'érodé du seuil de la fonction initiale (dans le cas bien sûr d'éléments structurants plans, seul envisagé ici). Il faudrait pour cela disposer de tous les seuils de l'image, ce qui ne manquerait pas de poser d'inextricables problèmes de taille mémoire et/ou de temps de traitement. De plus, des opérations plus complexes comme les amincissements ou épaisissements seraient beaucoup plus difficiles à réaliser. (Rappelons toutefois que cette méthode est

utilisée lorsqu'on est intéressé non pas par la totalité de l'image numérique transformée, mais seulement par un seuil défini à l'avance de cette image : les algorithmes de gradient ou de chapeau haut de forme fonctionnent ainsi).

- La troisième solution consiste à utiliser au mieux la représentation interne de l'image numérique. Cette représentation est constituée de plusieurs plans de bits stockés dans les mémoires binaires du processeur. Le problème consiste alors à expliciter les algorithmes permettant d'effectuer les transformations de base de la morphologie mathématique à partir des plans de bits. Or une analyse approfondie montre qu'il est très difficile de réaliser des opérations de voisinage (comme l'érosion, la dilatation, etc...) en partant des plans de bits de l'image numérique. On pourrait espérer réaliser une érosion numérique à l'aide d'une combinaison judicieuse d'opérations booléennes ou morphologiques sur les plans de bits de l'image. Cet espoir n'a malheureusement pas été comblé. Le seul cas où cela est possible est celui où l'élément structurant est constitué de deux points (adjacents ou non) : on est alors réduit à effectuer un inf entre deux valeurs (ou un sup dans le cas d'une dilatation).

Nous allons décrire les différentes opérations élémentaires effectuées entre deux pixels, en termes de transformations booléennes et morphologiques sur les plans de bits des images. Ces opérations sont au nombre de cinq : l'addition, la soustraction, l'inf (équivalent numérique d'une intersection booléenne) le sup (équivalent de l'union), l'inversion (équivalent de la complémentation).

#### DESCRIPTION DES OPERATIONS ELEMENTAIRES

Nous désignerons par les lettres A, B, etc..., des images numériques définies sur n bits.  $A_j, B_j$ , représentent le plan de bit j,  $j \in \{0, \dots, n-1\}$  des images. Un pixel x de l'image aura une valeur de gris p(x) (positif ou nul). A ce pixel correspondra sur chaque plan de bit j une valeur  $x_j \in \{0, 1\}$ . On a la relation :

$$p(x) = \sum_{j=0}^{n-1} 2^j x_j$$

Les algorithmes décrits ci-après fournissent les relations entre les plans de bits  $A_j$ ,  $B_j$  et  $C_j$  de trois images A, B et C liées par la relation :

$C = f(A, B)$ , f pouvant être une des cinq opérations citées plus haut.

Ces relations font toujours intervenir des plans intermédiaires que nous désignerons par les lettres  $M_j$ ,  $N_j$ . Ces plans définis pour chaque plan de bits permettent de transmettre aux plans supérieurs ou inférieurs (selon le cas) les éléments d'information nécessaires au calcul (retenues pour l'addition, masques pour l'inf, etc...).

#### 1/ L'addition et la soustraction

Soit l'image C, somme des images A et B.

$$C = A + B$$

On montre que le plan de bit  $C_j$  de l'image résultat est égale à :

(1)  $C_j = A_j * B_j * M_j$ , où l'opération \* représente la différence ensembliste symétrique de deux ensembles, et  $M_j$  un plan de retenue au niveau j. Ce plan est défini par la relation de récurrence suivante :

$$(2) \quad M_j = [A_{j-1} \cap B_{j-1}] \cup [A_{j-1} \cap M_{j-1}] \cup [B_{j-1} \cap M_{j-1}]$$

Cette formule montre que l'élaboration des plans de bits de l'image résultat se fait "en montant". Le plan de bit  $C_0$  doit être calculé d'abord, puis  $C_1$  etc...

Les relations ci-après s'obtiennent très facilement à partir des tables de vérité suivantes :

$A_j$	$B_j$	$M_{j-1}$	$C_j$	$M_j$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$M_{j-1}$  est le plan de retenue provenant des plans de bit inférieurs. On a bien évidemment  $M_{-1} = 0$  (plan de retenue initial).

Si l'ensemble  $M_{n-1}$  n'est pas vide, il y a débordement (la valeur obtenue est supérieure ou égale à  $2^n$ ). Dans ce cas, l'image C est égale à :

$$C = (A + B) \text{ mod } 2^n$$

Dans le cas de la soustraction, les formules obtenues sont les mêmes en remplaçant les plans  $B_j$  par leurs complémentaires  $\overline{B}_j$ .

$$C = A - B$$

$$(3) \quad C_j = A_j * \overline{B}_j * M_j$$

$$(4) \quad M_j = (A_{j-1} \cap \overline{B}_{j-1}) \cup (A_{j-1} \cap M_{j-1}) \cup (\overline{B}_{j-1} \cap M_{j-1})$$

Le plan  $M_{-1}$  doit être initialisé à 1.

Si  $M_{n-1}$  n'est pas égal à 1, il y a débordement (c'est-à-dire valeur négative). Dans ce cas encore, l'image C est égale à :

$$C = (A - B) \text{ mod } 2^n$$

La représentation des nombres négatifs est dite représentation en complément à 2.

## 2/ L'inversion d'image

L'inversion de l'image A produit une image C égale à :

$$C = (2^n - 1) - A$$

Les pixels de l'image A à zéro passent à  $2^n - 1$  et inversement. Considérant les équations précédentes, on a :

$$(5) \quad C_j = \bar{A}_j$$

## 3/ L'inf et le sup de deux images

Soit à calculer l'image :

$$C = \text{Inf}(A, B)$$

et donc les différents plans  $C_j$  qui lui correspondent. Pour cela, il nous faut introduire deux plans intermédiaires  $M_j$  et  $N_j$ . Ces deux plans, appelés plans de comparaison, indiquent au niveau  $j$  le sens de la relation d'ordre entre A et B. En effet, trouver l'inf des deux images, revient à déterminer pour chaque pixel  $x_A$  et  $x_B$  de l'image A et B lequel est inférieur à l'autre. Pour ce faire, la comparaison commence par les bits de poids forts et continue "en descendant" (à l'inverse de l'addition et de la soustraction). Au niveau  $j$ , l'un ou l'autre terme de l'alternative suivante est exact :

- Il n'est pas possible, d'après les valeurs des bits de niveau supérieur ou égal à  $j$  de déterminer laquelle des deux valeurs  $n_A$  ou  $n_B$  est l'inf.
- Il est possible de faire ce choix. Ce cas peut encore être subdivisé en deux parties :
  - l'inf est déterminé au niveau  $j$  par les valeurs respectives des bits  $x_{jA}$  et  $x_{jB}$ .
  - l'inf a déjà été déterminé à un niveau supérieur et dans ce cas, il est bien évident que les valeurs des bits  $x_{jA}$  et  $x_{jB}$  sont sans importance.

Toute cette analyse permet de définir le rôle des plans  $M_j$  et  $N_j$ .

$M_j$  indique, au niveau j, les pixels des images A et B tels que  $x_A \geq x_B$

$N_j$  indique, au niveau j, les pixels tels que  $x_A \leq x_B$ .

La restriction "au niveau j" doit être bien comprise. Elle signifie que seuls les plans de bit supérieurs ou égaux à j ont été analysés. L'exemple ci-dessous illustre la méthode :

	msb				lsb				(plan i)
A	0	1	1	0	1	1	0	0	( $x_A$ )
B	0	1	0	1	1	0	1	1	( $x_B$ )
M	1	1	1	1	1	1	1	1	
N	1	1	0	0	0	0	0	0	
		(a)	(b)				(c)		

(a) L'analyse des plans de bit 6 et 5 ne permet pas de savoir laquelle des deux valeurs  $x_A$  ou  $x_B$  est l'inf. M et N sont donc toujours égaux à 1.

(b) le plan de bit 4 détermine que  $x_B < x_A$ . M reste à 1, mais N passe à 0.

(c) Dès le basculement de M ou de N, les valeurs des bits de  $x_A$  et  $x_B$  des plans inférieurs à 4 est sans importance.

Les plans  $M_j$  et  $N_j$  se déterminent par les équations itératives suivantes :

$$(6) \quad M_j = M_{j+1} \cap (A_j \cup \bar{B}_j \cup \bar{N}_{j+1})$$

$$(7) \quad N_j = N_{j+1} \cap (\bar{A}_j \cup B_j \cup \bar{M}_{j+1})$$



Ces formules se retrouvent facilement. En effet, d'après ce qui a été dit plus haut,  $M_j$  est à 1 si et seulement si  $M_{j+1}$  est à 1 (sinon la condition  $\emptyset$  se propagerait) et si  $x_{jA} \geq x_{jB}$  (condition exprimée par  $A_j \cup \bar{B}_j$  au niveau des plans de bit) ou  $N_{j+1}$  est  $\emptyset$  (ce qui signifie que la détermination de la relation d'ordre entre  $x_A$  et  $x_B$  a déjà eu lieu dans les plans de bits supérieurs).

On a alors, si :

$$C = \text{Inf}(A, B)$$

$$(8) \quad C_j = (A_j \cap N_j) \cup (B_j \cap M_j)$$

De la même façon, si :

$$C = \text{Sup}(A, B)$$

$$(9) \quad C_j = (A_j \cap M_j) \cup (B_j \cap N_j)$$

La valeur initiale des plans de comparaison doit être égale à 1 :

$$M_n = N_n = 1$$

Quelques propriétés intéressantes peuvent être mises en évidence ; D'abord, on peut écrire :

$$(10) \quad A_j \cap M_j = A_j \cap M_{j+1}. \quad \text{En effet :}$$

$$A_j \cap M_j = A_j \cap M_{j+1} \cap (A_j \cup \bar{B}_j \cup \bar{N}_{j+1})$$

Le dernier terme contient  $A_j$ . On peut donc le supprimer et écrire la relation à démontrer.

De la même façon :

$$(10') \quad \begin{aligned} B_j \cap N_j &= B_j \cap N_{j+1} \\ A_j \cap \bar{M}_j &= A_j \cap \bar{M}_{j+1} \\ B_j \cap \bar{N}_j &= B_j \cap \bar{N}_{j+1} \end{aligned}$$

Une autre formule intéressante est la suivante :

$$(11) \quad M_j = M_{j+1} \cap (A_j \cup \bar{B}_j \cup \bar{N}_j)$$

ou bien encore :

$$(11') \quad N_j = N_{j+1} \cap (\bar{A}_j \cup B_j \cup \bar{M}_j)$$

L'intérêt de ces formules est de pouvoir économiser de la place mémoire lors des transformations comme nous le verrons plus tard.

#### LES TRANSFORMATIONS MORPHOLOGIQUES ELEMENTAIRES

On sait que les seules transformations réalisables de façon immédiate sont l'érosion (l'inf) et la dilatation (le sup) par un doublet de points. On sait également que cela est suffisant pour effectuer toutes les autres transformations. L'érosion hexagonale par exemple s'obtient grâce à trois érosions successives par des segments (c'est-à-dire des doublets de points adjacents).

##### 1/ Erosion, dilatation par un doublet de points

L'érosion de l'image A par un doublet de points distants de h est égale à :

$$A \ominus \left\{ \cdot \overset{h}{\longleftrightarrow} \cdot \right\} = \text{Inf}(A, A_{-h})$$

(l'origine de l'élément structurant est sur le point de gauche).

L'équation des masques devient :

$$(12) \quad M_j = M_{j+1} \cap (A_j \cup \bar{A}_{j-h} \cup \bar{N}_{j+1})$$
$$N_j = N_{j+1} \cap (\bar{A}_j \cup A_{j-h} \cup \bar{M}_{j+1})$$

Les relations  $A_j \cup \bar{A}_{j-h}$  et  $\bar{A}_j \cup A_{j-h}$  peuvent être réalisées par le biais de transformées morphologiques binaires. Ainsi :

$$A_j \cup A_{j-h} = A_j \oplus T, \text{ où } T = \cdot \emptyset$$

De la même façon :

$$A \oplus \{ \cdot \overset{h}{\leftrightarrow} \cdot \} = \text{Sup}(A, A_{-h})$$

## 2/ Erosion, dilatation hexagonale

Elles s'obtiennent par trois érosions (dilatations) par des doublets de points adjacents orientés dans les trois directions principales de la trame hexagonale. Cette façon de procéder est similaire à la procédure utilisée en binaire pour effectuer les opérations d'érosion et de dilatation hexagonales.

## 3/ Transformations euclidiennes et géodésiques

Pour les opérations de sup et inf, le fait de générer des plans de comparaisons  $M_j$  et  $N_j$  laisse la possibilité d'effectuer ces opérations à l'intérieur de masques définis par l'opérateur. En effet, soit à effectuer l'opération :

$C = \text{Sup}(A, B)$  pour tous les points de l'image contenus dans un masque  $D$ . En dehors du masque, plusieurs possibilités nous sont offertes :

$$\begin{aligned} \text{ou bien } C &= A \\ \text{ou } C &= B \\ \text{ou encore } C &= \emptyset \end{aligned}$$

Pour cela, il suffit d'initialiser les masques de comparaison  $M_n$  et  $N_n$  aux valeurs suivantes :

$$M_n = 1, N_n = D : C = \text{Sup}(A,B) \text{ dans } D \\ C = A \text{ à l'extérieur}$$

$$M_n = D, N_n = 1 : C = \text{Sup}(A,B) \text{ dans } D \\ C = B \text{ à l'extérieur}$$

$$M_n = D, N_n = D : C = \text{Sup}(A,B) \text{ dans } D \\ C = \emptyset \text{ à l'extérieur}$$

De la même façon, si l'opération à effectuer est :

$C = \text{Inf}(A,B)$  dans le masque  $D$ , les initialisations suivantes donneront les résultats énoncés ci-dessous :

$$M_n = 1, N_n = D : C = \text{Inf}(A,B) \text{ dans } D \\ C = B \text{ à l'extérieur}$$

$$M_n = D, N_n = 1 : C = \text{Inf}(A,B) \text{ dans } D \\ C = A \text{ à l'extérieur}$$

$$M_n = D, N_n = D : C = \text{Inf}(A,B) \text{ dans } D \\ C = 1 \text{ à l'extérieur}$$

La réalisation de transformations géodésiques est donc facile à mettre en oeuvre.

#### 4/ Épaississement et amincissement

Soit à effectuer l'épaississement de l'image  $A$  par un élément structurant  $T = (T_1, T_2)$

$$C = A \ominus T$$

La définition de l'épaississement fournit la valeur  $x_c$  de chaque pixel de l'image résultante suivant la formule :

$$x_c = \text{Inf}_{T_1}(x_A) \quad \text{ssi} \quad \text{Sup}_{T_2}(x_b) \leq x_A < \text{Inf}_{T_1}(x_A)$$

$$x_c = x_A \quad \text{sinon}$$

Compte tenu de ce qui a été dit plus haut, l'épaississement peut s'écrire comme le sup géodésique de l'image A et de son érodé par  $T_1$

$$C = A \ominus (T_1, T_2) = \{ \text{Sup}(A, A \ominus T_1) ; D \}$$

Le masque D est défini par l'ensemble des pixels tels que  $\text{Sup}_{T_2}(x_A) \leq x_A$ .

Or  $\text{Sup}_{T_2}(x_A) \in A \oplus T_2$ . Le masque D est donc constitué du plan de comparaison  $M_0$  obtenu à l'issue de l'opération :

$$\text{Sup}(A, A \oplus T_2)$$

Un processus similaire peut être défini pour l'amincissement :

$$C = A \circ T$$

$$x_c = \text{Sup}_{T_2}(x_A) \quad \text{ssi} \quad \text{Sup}_{T_2}(x_A) < x_A \leq \text{Inf}_{T_1}(x_A)$$

$$x_c = x_A \quad \text{sinon}$$

$$C = A \circ (T_1, T_2) = \{ \text{Inf}(A, A \oplus T_2) ; D \}$$

avec  $D = N_0$  à l'issue de l'opération :

$$\text{Inf}(A, A \oplus T_1)$$

#### LE PASSAGE DU NUMERIQUE AU BINAIRE

Il est donc clair, d'après ce qui a été dit plus haut, que les transformations morphologiques numériques sont réalisables par

le biais d'un processeur binaire manipulant les plans de bits de l'image. Ce codage de l'image par plans de bits est le plus simple possible, et le moins gourmand en mémoires.

Avant d'aborder les problèmes logiciels et matériels posés par une implantation réelle dans une machine comme le NS 1500, il convient de décrire succinctement les algorithmes permettant de générer les plans de bits de l'image, et inversement de restituer les niveaux de gris et les seuils à partir de l'image numérique.

### 1/ Codage de l'image

Un processeur binaire n'ayant généralement accès qu'aux différents seuils de l'image, la numérisation de cette dernière passe nécessairement par le balayage de ces différents seuils. Il faut alors s'assurer que les seuils successifs de l'image forment une suite monotone d'ensembles faute de quoi des erreurs de digitalisation des valeurs de gris peuvent se produire. L'amplitude de ces erreurs peut être de 100 %.

### 2/ Seuillage de l'image

Le seuillage de l'image pourrait être effectué niveau par niveau. Cette procédure longue peut être améliorée en utilisant un algorithme qui n'utilise au plus qu'une seule fois chacun des plans de bits.

On trouvera en annexe les algorithmes détaillés de ces deux transformations.

### IMPLANTATION DES ALGORITHMES SUR NS 1500

L'implantation des algorithmes précédemment décrits sur un analyseur type NS 1500 pose un certain nombre de problèmes d'adéquation des équations théoriques avec les possibilités réelles de la machine.

Les contraintes impératives à respecter sont un temps de traitement minimum ainsi qu'une occupation des mémoires-images raisonnable. Ces deux objectifs bien souvent incompatibles obligent à des prouesses de programmation assez acrobatiques, sur la plupart des machines binaires. On verra néanmoins qu'il est possible d'arriver à des temps de traitement raisonnables en utilisant au mieux les diverses propriétés mises en évidence lors de l'analyse des algorithmes.

### 1/ Implantation matérielle

Un point important doit être souligné dès à présent : l'impossibilité d'implanter les algorithmes sur images de gris avec la configuration actuelle et ceci pour plusieurs raisons :

- Le nombre de plans-images binaires est trop faible. Le NS 1500 dispose en effet de 8 plans binaires. Comme on a vu qu'il était nécessaire lors d'un traitement d'utiliser des plans intermédiaires en plus des plans de bits de l'image (que nous supposerons en nombre égal à 6, puisque le convertisseur analogique digital du NS 1500 travaille sur 64 niveaux de gris), cela laisse une place nulle pour le stockage des images numériques.

- Il n'y a pas actuellement de moyen simple de visualisation d'une image de gris sur le NS 1500.

Cependant, des extensions du NS 1500 sont envisageables. De plus, le NS 1500 étant piloté par un micro-ordinateur APPLE II, il est possible de pallier temporairement les faiblesses de la machine en mémoire et en visualisation, en utilisant quelques-unes des nombreuses extensions adaptables sur ce type de micro-ordinateur. Il existe par exemple des extensions mémoire de 64 K, 128 K et 256 K octets à des prix modiques qui permettent de porter la capacité mémoire à 1, 2 ou 5 images numérisées sur 6 bits.

Cette solution pénalise les vitesses de traitement et ne résout pas le problème de la visualisation, mais permet néanmoins

d'envisager quelques transformations.

La mémoire propre du NS 1500 se voit alors reléguer au rôle de mémoire de travail, rôle qui lui convient parfaitement puisque les processeurs l'accèdent directement.

## 2/ Architecture logicielle

D'après ce qui a été dit plus haut, les huit plans binaires du processeur constituent huit plans de travail. L'image à traiter doit être stockée entièrement ou en partie dans le NS 1500 avant d'effectuer l'opération. L'image résultat est alors transférée à nouveau dans les mémoires d'extension. Pour ne pas effectuer des transferts de façon systématique, on souhaite pouvoir stocker dans les mémoires de travail le maximum de plans binaires d'une image. Cette façon de faire permet d'effectuer une transformation hexagonale à l'aide de trois transformées linéaires sans avoir à sauvegarder à chaque étape l'image intermédiaire, faute de place. Deux plans de travail sont nécessaires pour le stockage des plans de comparaison  $M_j$  et  $N_j$ . Un plan supplémentaire est nécessaire pour leur calcul. Les équations (11) et (11') montrent que trois plans suffisent : en effet, on peut écraser  $N_{j+1}$  sans compromettre le calcul de  $M_j$  et vice-versa.

Lors des transformations comme l'érosion ou la dilatation par un doublet de points, les opérations  $A_{j-h}$  et  $\bar{A}_{j-h}$  peuvent être réalisées directement par le processeur du NS 1500. Pour des raisons d'occupation mémoire, il convient d'effectuer le calcul des plans de comparaison  $M_0$  et  $N_0$  finaux, avant de procéder au calcul des  $C_j$ . En effet, dans ce cas le plan  $N_0$  devient inutile et peut être utilisé comme plan de stockage intermédiaire. On peut écrire :

$$\begin{aligned}C &= \text{Sup}(A, B) \\C_j &= (A_j \cap M_0) \cup (B_j \cap M_0) \\C &= \text{Inf}(A, B) \\C_j &= (A_j \cap M_0) \cup (B_j \cap M_0)\end{aligned}$$



En effet  $M_0 = 1$  signifie  $A \geq B$ . Dans le premier cas, seuls les  $A_j$  sont pris en compte, et inversement seuls les  $B_j$  sont utilisés dans le calcul de l'inf.

De la même façon, la réalisation de transformations géométriques consiste uniquement à initialiser correctement les plans  $M_n$  et  $N_n$ .

### 3/ Vitesses d'exécution

Les équations énoncées dans les chapitres précédents permettent un calcul aisé des vitesses d'exécution des transformations. Rappelons qu'une transformation élémentaire du processeur NS 1500 est effectuée en 10 ms sur une image 280 x 192.

Les temps d'exécution des principales transformations sont résumés dans le tableau suivant :

Transformation	Nbre de passages	Temps
addition	49	490 ms
soustraction	id.	490 ms
inversion	6	60 ms
Inf	56	560 ms
Sup	56	560 ms
Erosion linéaire	53	530 ms
dilatation linéaire	id.	id.
Erosion hexagonale	159	1,59 s
dilatation hexagonale	159	1,59 s

Ces temps sont calculés pour une image 280 x 192 x 6 bits. L'image est supposée déjà présente dans les plans de travail. Les temps de transferts de ou vers la mémoire d'extension ne sont pas pris en compte. Pour transférer une image, il faut approximativement une seconde.

### CONCLUSIONS

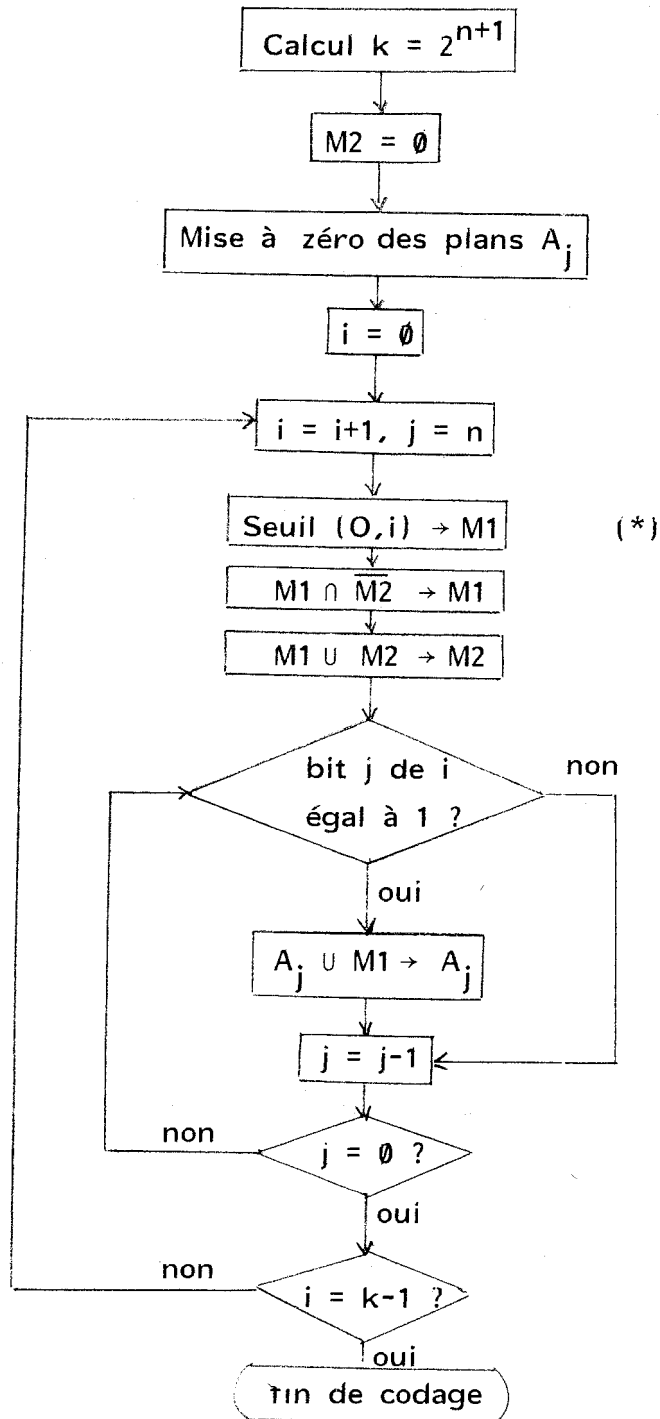
L'installation d'un logiciel de morphologie mathématique pour images de gris sur un processeur d'images comme le NS 1500 est réalisable et très avantageux ; l'investissement en matériel pour ce faire est peu onéreux. Les performances d'un tel logiciel le placent en bonne position en comparaison de logiciels comme MORPHOLOG exécutés sur un ordinateur du type VAX 11/750 (les temps CPU annoncés pour une érosion d'une image 256 x 256 x 8 bits sont de 1,5 à 2 s.). Si ces temps sont incompatibles avec des traitements de routine, ils sont très suffisants pour des logiciels ou des traitements de recherche. De plus, les transformations binaires se font à une vitesse incomparablement plus rapide que MORPHOLOG.

Toutes ces considérations font qu'il est intéressant de mettre au point des mémoires d'extension performantes sur ce type de matériel, associées à des circuits de visualisation à niveaux de gris.

Annexe A : ordinogrammes des opérations de codage et de seuillage d'une image numérique A.

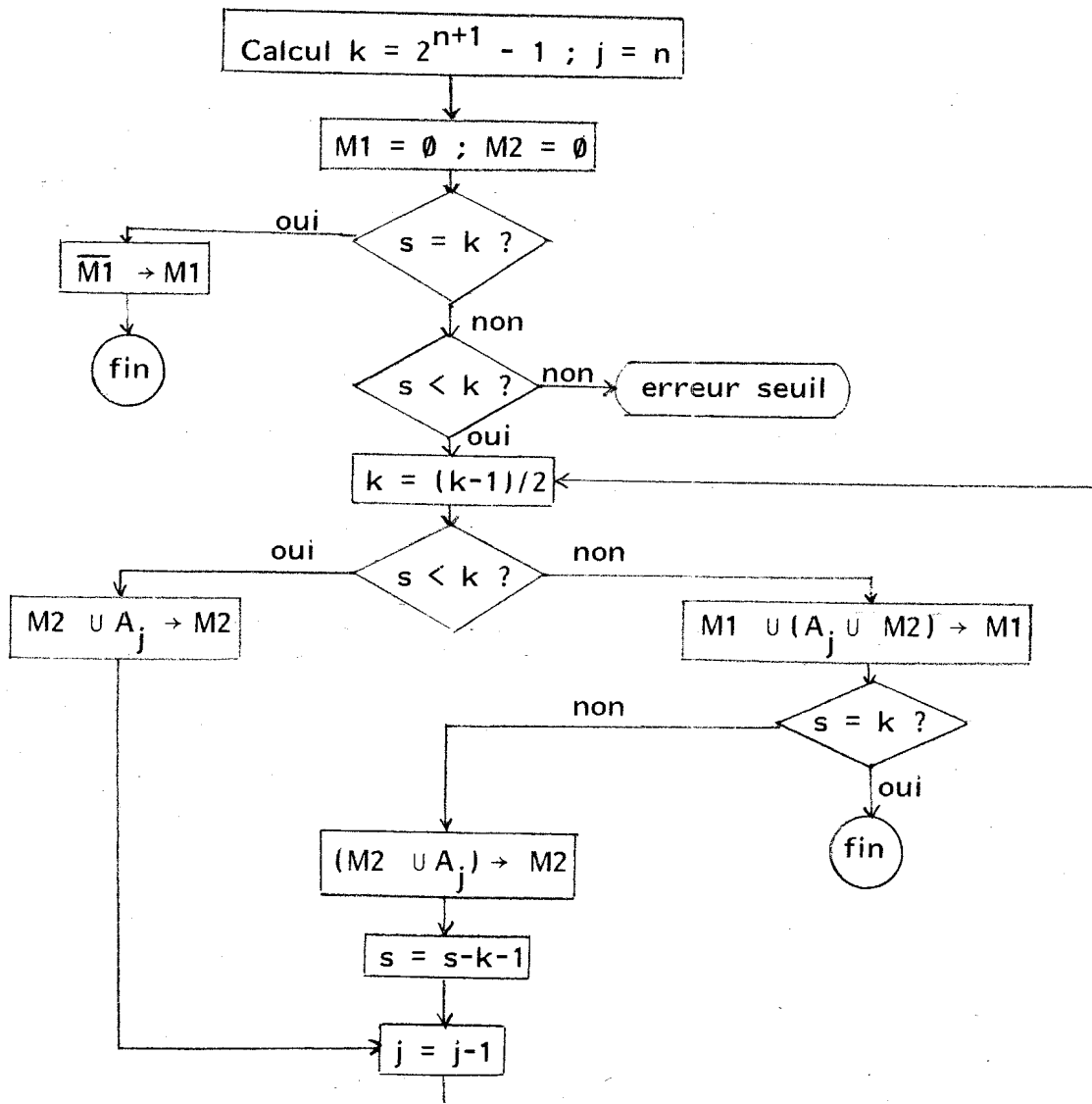
L'image est codée sur  $n+1$  bits. Deux plans de travail M1 et M2 sont utilisés

CODAGE



(\*) les opérations sont destinées à assurer l'inclusion des seuils les uns dans les autres.

SEUILLAGE



Le seuil final est dans le plan de travail  $M1$ .

Annexe B : L'érosion et la dilatation. Quelques précisions.

Le calcul de l'érosion d'une image A par un doublet de points peut être entreprise de la façon suivante (l'image est définie sur 6 bits) :

1/ Stockage des 6 bits  $A_j$  dans 6 plans-mémoire du NS 1500.

2/ Calcul de  $C_n$  (plan de bit de poids fort) de l'érodé :

$$C_n = A_n \ominus \{ . . \}$$

En effet, les formules (6), (7) et (8) donnent :

$$M_n = A_n \cup \bar{A}_{n-h}$$

$$N_n = \bar{A}_n \cup A_{n-h}$$

$$C_n = [A_n \cap (\bar{A}_n \cup A_{n-h})] \cup [A_{n-h} \cap (A_n \cup \bar{A}_{n-h})] = A_n \cap \bar{A}_{n-h}$$

ce qui est la définition même de l'érodé.

3/ Les plans  $M_n$  et  $N_n$  peuvent être calculés à l'aide des formules données plus haut. Le plan  $A_n$  désormais inutile peut être libéré, et servir au calcul des plans  $M_j$  et  $N_j$  successifs.

Il est plus avantageux d'utiliser les plans de comparaison complémentés. En effet :

$$P_j = \bar{M}_j = \bar{M}_{j+1} \cup (\bar{A}_j \cap B_j \cap N_{j+1}) = P_{j+1} \cup (\bar{A}_j \cap B_j \cap \bar{Q}_{j+1})$$

$$Q_j = Q_{j+1} \cup (A_j \cap \bar{B}_j \cap \bar{P}_{j+1})$$

Dans le cas des érosions et des dilatations, le calcul de  $\bar{A}_j \cap A_{j-h}$  et  $A_j \cap \bar{A}_{j-h}$  peut s'effectuer en une passe. En effet :

$\bar{A}_j \cap A_{j-h} = A_j * \{0\ 1\}$ , transformée en tout ou rien programmable facilement dans un des processeurs du NS 1500.

De la même façon :

$$A_j \cap \bar{A}_{j-h} = A_{j-h} * \{0\ 1\}.$$

Le recentrage des transformations peut être effectué à l'aide du second processeur. Le positionnement relatif des plans  $P_j$  et  $Q_j$  par rapport au plan  $A_j$ , permet également, s'il est contrôlé, d'accélérer certaines transformations. Ainsi, dans le cas de la dilatation linéaire, l'opération :

$C_j = (A_j \cap \bar{P}_j) \cup (A_{j-h} \cap \bar{Q}_j)$  peut s'effectuer plus rapidement (3 passes au lieu de 4) si les points du plan  $Q_j$  sont en vis-à-vis des points de  $A_{j-h}$ . Le deuxième membre se calcule alors en un passage, à l'aide des deux processeurs.