

50376  
1989  
217

50376  
1989  
217

N° d'ordre 378

**THESE**

présentée à

**L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE  
LILLE FLANDRES ARTOIS**

pour l'obtention du titre de

**DOCTEUR**

Spécialité: Productique, Automatique et  
Informatique Industrielle

par

**Pierre-Marie TANCREZ**



**UN SYSTEME DE VISION LINEAIRE POUR L'INSPECTION  
ET LA CARACTERISATION EN TEMPS REEL DE PRODUITS PLATS  
EN DEFILEMENT CONTINU**

Soutenue le 28 JUIN 1989



50376  
1989  
217

50376  
1989  
217

## AVANT-PROPOS

Les recherches présentées dans ce mémoire ont été réalisées dans le cadre du développement de l'atelier flexible de confection du Centre d'Automatique de Lille animé par M. le Professeur P. VIDAL. Je le remercie pour l'accueil qu'il m'a réservé au sein du laboratoire et pour l'honneur qu'il m'a fait en acceptant d'être, à la fois, le Président de la commission d'examen et rapporteur.

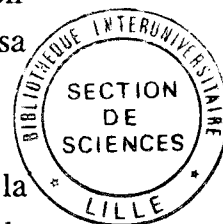
Je remercie vivement M. le Professeur M. BOURTON de l'Université de Valenciennes pour avoir accepté de juger le contenu de ce mémoire et pour sa présence parmi les membres de ce jury en tant que rapporteur.

Je suis gré à M. le Professeur J.G. POSTAIRE pour la confiance qu'il m'a accordée en dirigeant mes recherches. Je le remercie particulièrement pour l'aide constante qu'il m'a apportée, ses conseils et sa grande disponibilité.

Je tiens à exprimer ma profonde reconnaissance à M. P. BONNET, Maître de Conférence au Centre d'Automatique, pour ses conseils et le remercie vivement pour sa présence à la commission d'examen comme examinateur.

Que M. M. GUEDJ de la Direction des Etudes et Recherches de l'EDF à Chatou trouve ici l'expression de ma profonde gratitude pour avoir bien voulu porté un jugement sur ce travail et mes remerciements pour sa participation au jury.

Que M. F. GERMAIN, Responsable du Département Vision de la Société Digital Design, qui a accepté de juger le contenu de mémoire et de participer à la commission d'examen trouve ici l'expression de mes sincères remerciements.



Je ne saurais terminer cet avant-propos sans remercier tous les membres du Laboratoire d'Automatique avec qui j'ai eu la chance de travailler et qui ont toujours su faire régner la bonne ambiance.

## **CHAPITRE I**

### **INSPECTION AUTOMATIQUE DE PRODUITS PLATS EN DEFILEMENT CONTINU.**

#### **I. INTRODUCTION.**

Le contrôle visuel, à différents niveaux dans les chaînes de production, est certainement l'un des moyens les plus efficaces pour éliminer les produits défectueux au cours d'un processus de fabrication. En effet, si l'on confie les contrôles visuels à des opérateurs humains, on court le risque de ne pas détecter toutes les anomalies. L'opérateur humain, qui peut être excellent sur de courts intervalles de temps, se révèle en général peu fiable sur de longues périodes, compte tenu de la monotonie de la tâche d'inspection et de la fatigue visuelle qu'elle engendre. Il en résulte que, les décisions prises par les opérateurs humains sont fonction :

- de leur état de fatigue,
- de leurs qualités de perception, et
- de leur subjectivité.

Ce dernier aspect peut être plus ou moins maîtrisé si l'on procède à des périodes d'apprentissage et de contrôle périodique des performances des opérateurs. Mais le coût de ces recyclages est tel qu'ils ne sont que très rarement pratiqués.

Le contrôle visuel est donc souvent un des maillons faibles d'un système de production automatisé. Mais aujourd'hui, l'intégration de machines de vision au niveau du contrôle qualité permet de pallier les défaillances des contrôleurs. Bien qu'en général, les performances d'un système de vision soient moins bonnes que celles d'un opérateur humain sur de brèves périodes, on constate que l'apport des systèmes d'inspection automatique se révèle d'autant plus important que le temps d'inspection augmente [HOL-84], [ARN-88]. De plus, le contrôle visuel, lorsqu'il est confié à des machines, permet de garantir des performances constantes et des prises de décisions objectives.

## II. IMPORTANCE DU SYSTEME D'ECLAIRAGE.

Tout praticien en Visionique connaît l'importance des problèmes liés à la conception d'un système d'éclairage. En effet, les meilleures caméras et les meilleurs logiciels ne permettent de capter et d'exploiter que l'information visuelle présente dans la scène observée. Mal éclairée, cette dernière ne révélera pas tout son contenu.

Il est déjà très difficile d'obtenir des conditions d'éclairage satisfaisantes en laboratoire. En effet, le flux lumineux délivré par les sources de lumière est rarement homogène. L'intensité lumineuse a tendance à décroître au fur et à mesure que l'on s'éloigne de l'axe d'une lampe et cette décroissance n'est pas toujours la même dans toutes les directions. On sait que l'on peut compenser cette non-uniformité par l'emploi d'un assemblage de plusieurs sources lumineuses. Mais, même si l'oeil humain a l'impression d'être en présence d'un flux homogène, l'analyse de l'intensité lumineuse reçue par la scène met généralement en évidence de fortes variations spatiales. L'emploi de réflecteurs augmente l'intensité lumineuse reçue par la scène, mais n'améliore que rarement l'uniformité des flux.

Dans certaines configurations, la grande largeur de la scène à observer pose également des problèmes de réalisation du dispositif d'éclairage, que l'inspection s'effectue par vision directe ou par transparence. En effet, il est très difficile de construire des lignes isolux de plus de 1,50 mètre de longueur, compte tenu de la taille des tubes lumineux proposés par

les fabricants. On est contraint d'assembler des tubes ou des sources lumineuses ponctuelles qui, malgré l'emploi de réflecteurs, ne donnent qu'une approximation d'une ligne à éclairage uniforme.

Il serait d'ailleurs très intéressant de développer des logiciels de C.A.O. de systèmes d'éclairage qui permettraient d'optimiser le positionnement des sources lumineuses et des réflecteurs. Des chercheurs du Centre d'Automatique de l'Université de Lille s'y emploient actuellement.

D'autres problèmes d'éclairage apparaissent lorsque l'on passe de l'environnement contrôlé et protégé du laboratoire à l'environnement moins contrôlable de l'atelier de production. Des poussières, de la fumée ou de la vapeur peuvent s'interposer entre l'objectif de la caméra et la scène à observer. Il arrive que des ombres ou des sources de lumière parasites viennent interférer avec le dispositif d'éclairage et les lampes peuvent vieillir de manière non uniforme.

Pour conserver les performances d'un système de vision sans procéder à des investissements lourds et parfois difficiles à installer tels que des tunnels, et sans être astreint à une maintenance délicate et permanente, il est impératif de développer des procédures d'analyse d'images aussi peu sensibles que possible aux variations spatiales et temporelles des conditions d'éclairage.

Seule la mise à la disposition des industriels de logiciels robustes par rapport aux conditions d'éclairage permettra d'envisager une implantation simple des systèmes de vision dans les processus de production, et de ce fait, augmentera leur champ d'application.

### **III. INSPECTION AUTOMATIQUE PAR VISION LINEAIRE.**

C'est par ce problème de robustesse par rapport aux variations spatiales et temporelles de l'éclairage que nous abordons le travail présenté dans ce mémoire. Cette étude se situe dans le cadre de l'inspection de produits plats en défilement continu sur des convoyeurs. L'inspection des matériaux plans présente un très grand intérêt dans le textile, l'habillement,

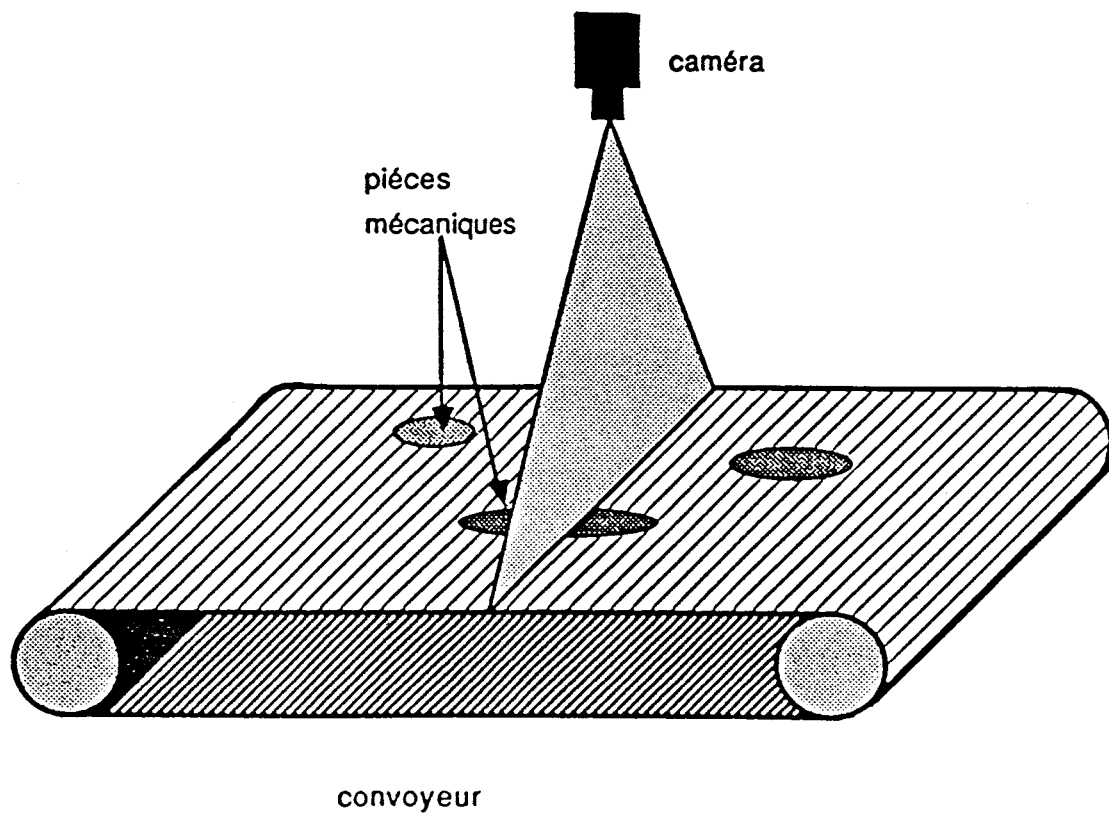
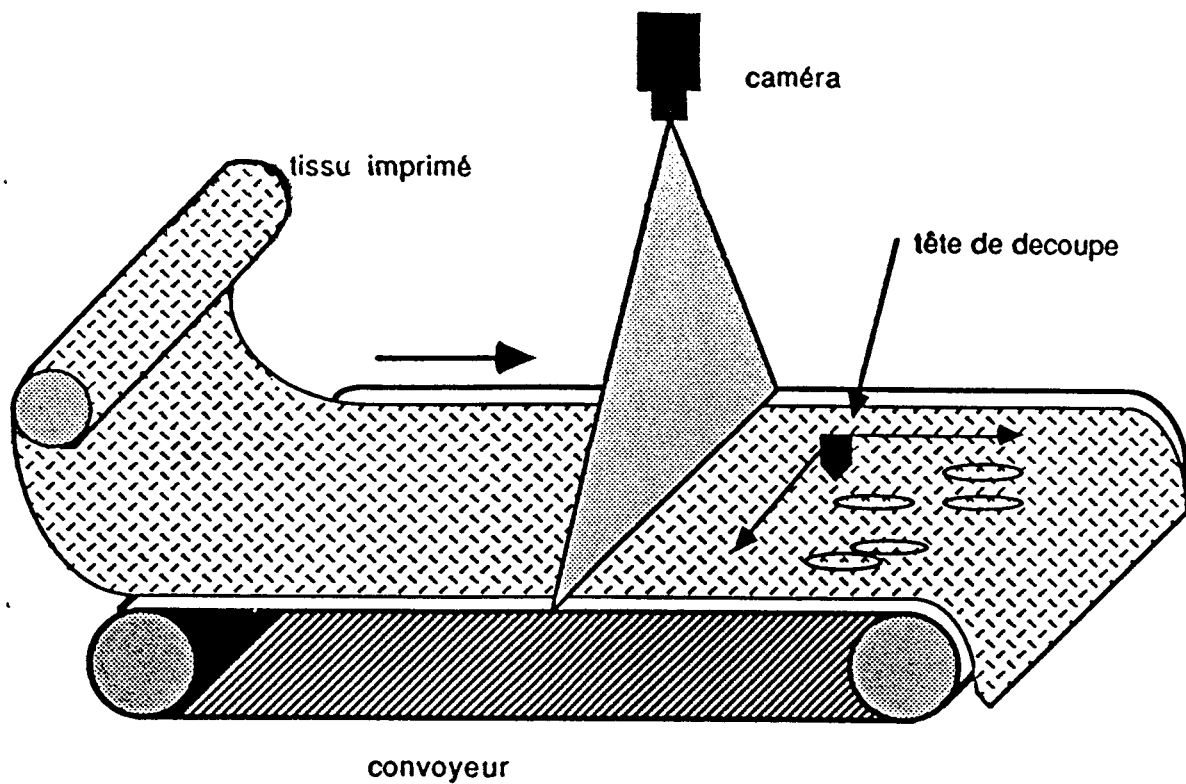
la sidérurgie, l'industrie des matières plastiques, la papeterie, etc... . L'inspection d'objets en défilement sur des convoyeurs concerne de très nombreux secteurs de l'industrie manufacturière.

L'inspection de produits plats en défilement s'effectue, en général, en utilisant des caméras linéaires. On profite du mouvement des produits à inspecter sous la caméra dans la direction perpendiculaire à la ligne des éléments photosensibles pour reconstituer, ligne par ligne, la scène en mouvement observée (figures 1. a et b).

A intervalles de temps réguliers, la caméra linéaire acquiert une image sous forme d'une ligne. Le signal vidéo ainsi obtenu est discrétisé en un signal numérique constitué d'un nombre  $N$  d'éléments égale au nombre d'éléments photosensibles qui composent la barrette de la caméra. Ainsi, à chaque diode est associée une valeur correspondant au niveau de gris moyen que celle-ci a intégré pendant le temps de pose. Cette valeur peut avoir une dynamique sur 64 ou 256 niveaux de gris suivant que le codage effectué au niveau du convertisseur analogique-numérique est sur 6 ou 8 bits. Certains systèmes de vision permettent des codages sur un nombre de bits plus important. Nous obtenons donc, en sortie de caméra, un signal discrétisé constitué des  $N$  valeurs successives des niveaux de gris reçus par les  $N$  éléments photosensibles de la barrette.

Notons que pour éviter les aléas au niveau des régimes transitoires entre les lignes successives acquises par le système de vision, on ignore généralement les valeurs des niveaux de gris captés par les éléments photosensibles des extrémités de la barrette.

De ce fait, nous ne tenons pas compte des niveaux de gris des diodes dont l'abscisse est en-deçà d'une certaine borne inférieure, notée  $X_{inf}$  correspondant à l'abscisse de la première diode, dans le sens des abscisses croissantes, dont le niveau de gris est considéré comme étant fiable, et au delà d'une borne supérieure, notée  $X_{sup}$  correspondant à l'abscisse de la dernière diode dont le niveau de gris est fiable. Ainsi, dans tout traitement, l'abscisse  $x$  d'une diode est comprise dans l'intervalle  $[X_{inf}, X_{sup}]$ .



figures 1.a et 1.b : deux exemples d'inspection automatique de matériau en défilement continu sous une caméra linéaire.



Ce signal numérique de la scène observé finalement obtenu est appelé, dans la suite du manuscrit, l'image-ligne.( cf figure 2 )

La grande résolution des caméras linéaires qui, dans les versions standards atteignent 4096 éléments photosensibles, permet d'obtenir une finesse d'analyse dans le sens transversal beaucoup plus importante que celle résultant de l'utilisation de caméras matricielles. Dans le sens longitudinal, la résolution n'est limitée, pour une vitesse de défilement donnée du produit à inspecter, que par la cadence d'acquisition des image-lignes. En effet, le temps s'écoulant entre deux acquisitions doit permettre de mémoriser le contenu visuel de la ligne et d'effectuer tous les traitements nécessaires à l'application envisagée. Ici, apparaît l'aspect temps réel de l'étude. En effet, plus vite sera analysée chaque ligne saisie par le système de vision, meilleure sera la résolution longitudinale du procédé d'inspection.

#### **IV. AMELIORATION DE LA ROBUSTESSE PAR RAPPORT AUX VARIATIONS D'ECLAIRAGE.**

Compte tenu des conditions d'exploitation évoquées ci-dessus, il apparaît impératif d'implanter sur les systèmes de vision linéaire des procédures d'analyse d'images aussi peu sensibles que possible aux variations, tant spatiales que temporelles, des conditions d'éclairage.

Un bref rappel des différentes procédures de segmentation utilisées en vision linéaire nous permet de dégager, au chapitre II, les limites des procédures de bas niveau orientées temps réel.

Pour pallier les faiblesses de ces procédures nous proposons, au chapitre III, un nouvel opérateur de segmentation en vision linéaire, faisant appel à des concepts de Morphologie Mathématique en niveaux de gris.

Les formes présentes dans la scène observée sont alors extraites de l'image par une procédure d'étiquetage dynamique qui ne nécessite que la mémorisation de deux lignes consécutives. Le système d'inspection par vision linéaire développé au cours de cette étude est complété, au chapitre IV, par

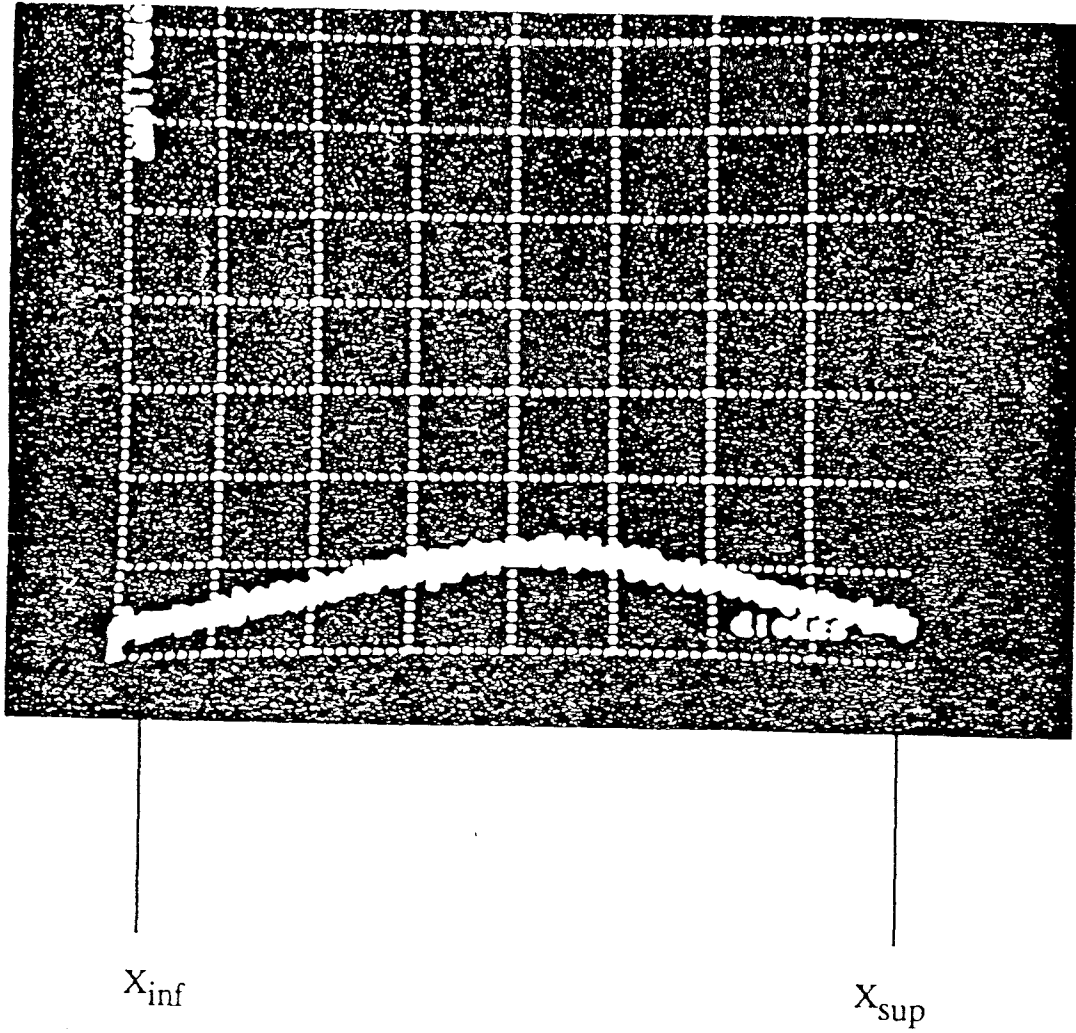


figure 2 : une image-ligne.

la détermination d'attributs morphométriques en temps réel, c'est à dire à la cadence d'acquisition des image-lignes.

Enfin, l'implantation temps réel de cette approche sur une architecture multi-processeurs est présentée dans le dernier chapitre.

## CHAPITRE II

### SEGMENTATION EN VISION LINEAIRE.

Notre propos, dans ce chapitre, est de présenter, de façon brève et concise, les techniques de base du traitement des images numériques recueillies par un système à caméra linéaire.

On peut regrouper les différentes procédures de segmentation de bas niveau existantes en trois types de méthode, selon que l'on fait appel à :

- des techniques de seuillage,
- des techniques soustractives,
- des opérateurs différentiels.

#### I. TECHNIQUES DE SEUILLAGE.

Le seuillage du signal délivré par une caméra linéaire est une opération simple dans la mesure où le seuil est constant sur toute l'image-ligne, c'est à dire pour l'ensemble des éléments photosensibles. (cf figure 3).

Le résultat de cette opération de seuillage peut être mémorisé sous la forme des numéros des diodes où le signal vidéo franchit le seuil, dans un sens comme dans l'autre. Ce seuillage peut être réalisé soit par logiciel, soit par des opérateurs câblés. Dans ce dernier cas, le seuillage peut être itéré en modifiant le seuil et en réinjectant l'image-ligne dans le dispositif de

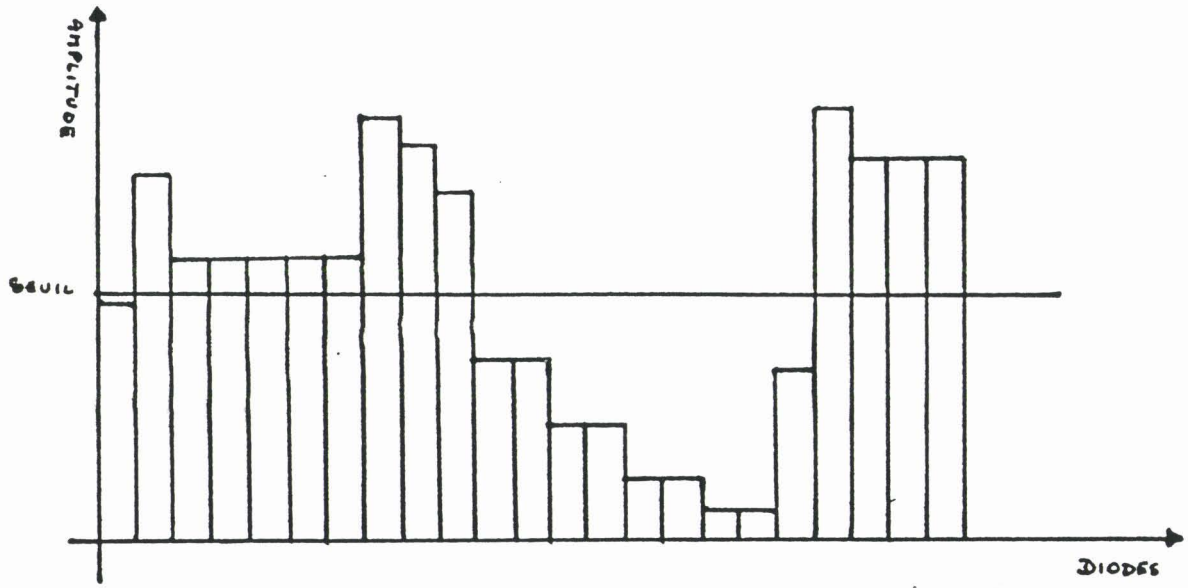


figure 3 : le seuillage simple d'image-ligne.

seuillage. Cette méthode est très intéressante car elle permet le multi-seuillage, avec des temps de traitement très brefs.

Comme pour toutes les méthodes de seuillage, le succès de ces procédures est conditionné par l'ajustement du seuil. Ce dernier peut être confié à un opérateur humain dans des cas simples, par exemple quand des pièces en défilement présentent un fort contraste avec le convoyeur qui les supporte et quand l'éclairage est bien uniforme.

Mais lorsque, pour les raisons évoquées au chapitre précédent, l'éclairage présente des variations spatiales le long des image-lignes, la procédure de seuillage peut être vouée à l'échec. La figure 4 illustre une situation type où la composante "éclairage" du signal n'est pas uniforme et ne permet pas de retrouver les objets par simple seuillage.

Pour le type d'image-ligne de la figure , où l'éclairage n'est pas constant, on a généralement recours à une technique de soustraction d'une image-ligne dépourvue d'objet et constituant l'image de référence, appelée "fond" de la scène. Ce signal peut être obtenu au cours d'une phase d'apprentissage qui consiste à mettre en mémoire la luminance d'une image-ligne de la scène dépourvue des objets que l'on doit en extraire [VIS-87].

## II. TECHNIQUES SOUSTRACTIVES.

La technique consiste à soustraire, point par point, le signal de référence, représentant le "fond" de la scène et mémorisée dans la "mémoire de fond", au signal représentant l'image-ligne courante.

Cette procédure n'est applicable que lorsque l'on peut disposer du fond de la scène sans objet, ce qui n'est pas toujours le cas. De plus, si la luminance du fond varie dans le temps, cette image-ligne de référence doit être réactualisée. On comprend aisément qu'il n'est pas toujours possible de disposer du fond en cours de production pour effectuer cette réactualisation.

Toutefois, sous certaines hypothèses restrictives sur la position des objets, il est possible de déterminer un modèle polynomial de la luminance

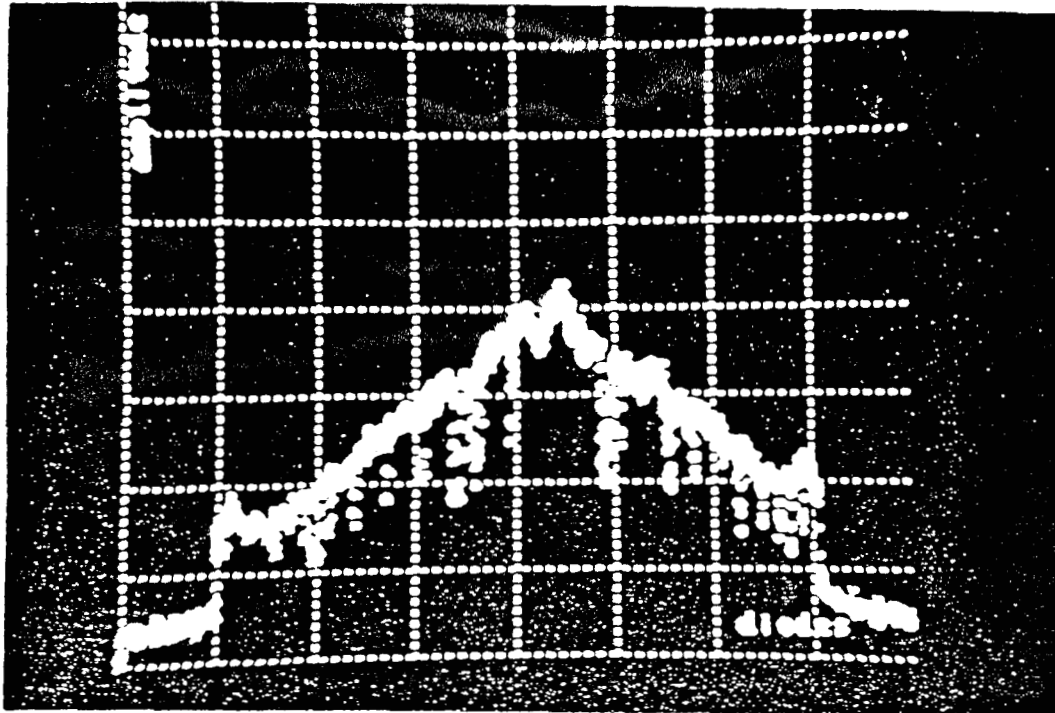


figure 4 : la non-uniformité de la composante "éclairage" le long d'une image-ligne

du fond, même si celui-ci n'est pas entièrement dépourvu d'objet [DEL-87]. Cette approche qui permet d'éliminer la composante "éclairage" n'est en fait utilisable que si les objets sont de petite taille et ne recouvrent qu'une faible partie du champ de vision. De plus, le temps de calcul nécessaire à cette modélisation interdit son utilisation dans une optique temps réel.

Plus récemment, une amélioration de cette approche a été proposée afin de s'affranchir des hypothèses restrictives sur la position, la taille et le nombre d'objets dans le champ de la caméra [DOU-88]. Les temps de calcul associés à cette technique, encore plus importants, interdisent complètement l'utilisation de cette méthode en temps réel.

Comme pour l'analyse des images numériques matricielles, une autre approche pour s'affranchir des variations spatiales et temporelles de l'éclairage consiste à faire appel à des opérateurs différentiels pour détecter les contours présents dans l'image.

### III. OPERATEURS DIFFERENTIELS.

Deux types d'opérateurs différentiels permettent de segmenter une image sur des critères de variations locales de l'intensité lumineuse : ce sont les opérateurs de type gradient et de type laplacien.

Leur utilisation permet de détecter les contours présents dans des images non homogènes, où les techniques de seuillage et de soustraction du fond ne permettent pas une segmentation satisfaisante. En effet, les variations locales d'intensité au niveau du contour d'un objet sont généralement beaucoup plus importantes que celles de l'éclairage le long d'une image-ligne ou entre des image-lignes consécutives, ce qui permet de détecter les contours de manière fiable.



### III.1. Opérateurs de type "gradient".

Soit le point de coordonnées  $(x,y)$  dans une image bidimensionnelle et  $f(x,y)$ , la valeur du niveau de gris en ce point. Le gradient de cette fonction de niveau de gris en  $(x,y)$  est :

$$\text{GRAD}[f(x,y)] = \frac{\partial f(x,y)}{\partial x} \vec{u} + \frac{\partial f(x,y)}{\partial y} \vec{v}$$

où  $\vec{u}$  et  $\vec{v}$  sont les vecteurs unitaires du repère orthonormé lié à l'image, l'ordonnée  $y$  représentant le sens du déplacement de la scène sous la caméra et l'abscisse  $x$ , le sens transversal dans la direction des image-lignes recueillies par la caméra.

#### III.1.1. Le gradient unidimensionnel.

Si on se limite à l'analyse d'une seule image-ligne saisie par une caméra linéaire, le gradient se réduit à :

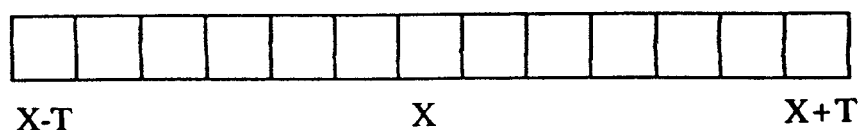
$$\text{GRAD}[f(x)] = \frac{\partial f(x)}{\partial x} \vec{u}$$

où  $\vec{u}$  est le vecteur unitaire du repère lié à cette image-ligne.

Dans le cas discret, ce gradient est approché par une différence de niveaux de gris des points voisins :

$$\text{GRAD}[f(x)] = \frac{f(x+T) - f(x-T)}{(x+T) - (x-T)}$$

où l'entier  $T$  définit la largeur de la fenêtre sur laquelle le gradient est calculé, c'est à dire le nombre de points de part et d'autre du point  $x$  considéré.



Le choix de la taille  $T$  de la fenêtre de calcul du gradient est très important. Si la fenêtre est trop petite, le gradient est sensible aux petites variations locales non significatives de l'intensité lumineuse. Par contre, si elle est trop grande, l'opérateur est beaucoup moins sensible aux variations significatives du niveau de gris et la détection est moins performante.

Notons que pour explorer une image-ligne, on peut choisir un pas de progression  $k$  de la fenêtre le long de l'image-ligne, différent de l'unité. Une valeur importante de ce pas d'échantillonnage augmentera la vitesse de traitement de l'image-ligne mais au détriment de la qualité de la détection des contours. Un compromis doit être trouvé dans le choix de ces deux paramètres afin d'allier qualité et rapidité.

Soit une image-ligne délimitée par un bord inférieur dont l'abscisse est notée  $X_{\text{inf}}$ , et par un bord supérieur dont l'abscisse est notée  $X_{\text{sup}}$ , pour un pas d'échantillonnage  $k$  et une taille de fenêtre  $T$ , le résultat de l'opérateur gradient sur la fonction  $f(x)$  représentant le niveau de gris de la diode d'abscisse  $x$  est :

$$\text{GRAD}[f(x+\alpha k)] = \frac{f(x+\alpha k+T) - f(x+\alpha k-T)}{2T}$$

avec  $\alpha$  variant de 0 à  $(X_{\text{inf}} - X_{\text{sup}} - 2T) / k$

L'implantation logicielle de cet opérateur ne présente aucune difficulté sur un système de vision linéaire et la mise au point est aisée grâce à la possibilité de paramétrer la taille  $T$  de la fenêtre et le pas d'échantillonnage  $k$ . La mise en oeuvre de cet opérateur est très proche de celle des techniques de seuillage : seules les positions où la réponse de l'opérateur franchit un seuil donné, dans un sens comme dans l'autre, sont mises en mémoire, ligne après ligne.

Mais ce gradient unidimensionnel, certes efficace et rapide, ne peut pas, par sa nature même, détecter les variations d'intensité lumineuse intervenant entre deux image-lignes successives. Cette limitation engendre de gros problèmes à la phase d'extraction de contours car nous ne pouvons détecter les variations significatives qui peuvent se produire entre des image-lignes consécutives. De ce fait, il nous faut envisager l'implantation d'un opérateur gradient bidimensionnel permettant de détecter les variations de l'intensité à la fois sur une image-ligne et entre deux ou plusieurs image-lignes consécutives. Nous retrouvons, ici, le concept du gradient utilisé dans l'analyse des images matricielles.

### III.1.2. Le gradient bidimensionnel.

Le gradient bidimensionnel est calculé à l'aide de deux convolutions: l'une, notée  $G_x$ , est utilisée pour estimer le gradient dans la direction transversale, c'est à dire le long des image-lignes et l'autre,  $G_y$ , pour estimer le gradient dans la direction longitudinale, c'est à dire celle du déplacement. Les tailles des fenêtres de convolution les plus couramment utilisées, identiques pour les deux convolutions, sont de  $3 \times 3$ ,  $5 \times 5$  ou  $7 \times 7$ .

Parmi les opérateurs de convolution les plus usuels, citons ceux définis par les masques suivants :

$$G_x = \begin{vmatrix} 1 & 0 & -1 \\ \beta & 0 & -\beta \\ 1 & 0 & -1 \end{vmatrix}$$

$$G_y = \begin{vmatrix} -1 & -\beta & -1 \\ 0 & 0 & 0 \\ 1 & \beta & 1 \end{vmatrix}$$

Si  $\beta = 0$  nous obtenons l'opérateur de Sobel [BAS-79].

Si  $\beta = 1$  nous obtenons l'opérateur de Prewitt [PRE-66].

Le module du gradient, noté  $G$ , est ensuite calculé sous la forme d'une combinaison linéaire des valeurs absolues de  $G_x$  et  $G_y$  :

$$G = a \cdot |G_x| + b \cdot |G_y|$$

où  $a$  et  $b$  prennent les valeurs 0, 1, 2 ou 4 selon les directions que l'on désire privilégier.

Le seuillage de la réponse de cet opérateur permet de détecter les zones de l'image où l'intensité lumineuse varie brusquement, indépendamment du niveau lumineux proprement dit. Notons que les contours ainsi obtenus, dont l'épaisseur peut être importante, ne sont pas toujours fermés et parfois perturbés par du bruit difficile à éliminer.

La mise en oeuvre de ces opérateurs nécessite la mémorisation de 3, 5 ou 7 image-lignes consécutives selon la taille des masques utilisés, et la manipulation de 9, 25 ou 49 éléments dans la fenêtre de calcul. Une telle procédure ne peut être envisagée au niveau d'une implantation purement logicielle car cette solution, très coûteuse en temps de calcul, nous éloignerait de notre objectif temps réel.

Certains systèmes proposent des solutions matérielles grâce à l'emploi d'opérateurs câblés. Si les performances temporelles de ces opérateurs répondent parfaitement aux impératifs de temps réel, l'expérience montre que leur mise en oeuvre demeure délicate car elle nécessite un réglage très fin de l'éclairage et un ajustement rigoureux du seuillage. [DUP-89].

### III.2. Opérateur de type "laplacien".

Le laplacien peut être défini comme la dérivée seconde de l'intensité lumineuse.

Au point de coordonnées  $(x,y)$  d'une image bidimensionnelle, le laplacien de la fonction de niveau de gris  $f(x,y)$  est donné par :

$$\text{LAPL}[f(x,y)] = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

Comme nous l'avons fait pour le gradient, nous pouvons différencier deux cas : le laplacien unidimensionnel et le laplacien bidimensionnel.

### III.2.1. Laplacien unidimensionnel.

Le laplacien unidimensionnel peut s'exprimer aisément à partir de la formulation du gradient unidimensionnel précisé au paragraphe précédent. Il vient :

$$\text{LAPL}[f(x)] = \text{GRAD}[f(x+1)] - \text{GRAD}[f(x)].$$

La mise en oeuvre de cet opérateur unidimensionnel repose donc sur celle du gradient unidimensionnel.

### III.2.2. Laplacien bidimensionnel.

Des approximations discrètes de cet opérateur sont données par les masques [ROS-82], [SAM-84] :

$$\begin{vmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{vmatrix}$$

ou

$$\begin{vmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$

Cet opérateur différentiel a des coefficients constants et homogènes du second ordre. Il introduit des valeurs positives et des valeurs négatives. Les passages à la valeur zéro localisent les points d'inflexion de la fonction de niveau de gris  $f(x,y)$ . Les contours des formes présentes dans l'image sont ainsi détectés. Le principal avantage du laplacien est son isotropie. Mais le laplacien, plus que le gradient, favorise les points isolés dus au bruit, ainsi que les lignes fines et les extrémités [COS-85], [LEV-81].

D'autres opérateurs différentiels ont été développés afin de garantir une meilleure immunité au bruit, en augmentant la taille des fenêtres de calculs mais cela évidemment au détriment du temps de calcul. [KUN-87], [NAL-86].

### III.3. Conclusion.

En fait, que l'on utilise un opérateur différentiel de type gradient ou de type laplacien, la détection des contours des objets dans une image pose un problème délicat.

Dans des cas simples, le seuillage de la réponse de l'opérateur permet de détecter les contours recherchés. Mais l'image différentielle est souvent bruitée et la réponse de l'opérateur n'est pas constante le long des contours. Dans ces conditions, il est difficile de détecter des contours fermés significatifs. Un seuil trop haut ne permet de retenir que des bribes de contours. A l'inverse, un seuil trop bas donne des contours épais, noyés dans le bruit.

## IV. CONCLUSION.

Ce bref exposé des principales techniques de segmentation, actuellement existantes pour les systèmes de vision à caméra linéaire, montre les limites des méthodes de seuillage, de soustraction de fond et de détection des contours par des opérateurs différentiels.

Aucune de ces méthodes ne résiste aux exigences de l'inspection automatique de produits en défilement continu observés par une ou plusieurs

caméras linéaires, à savoir : rapidité et robustesse par rapport aux variations spatiales et temporelles du système d'éclairage.

Cette constatation nous incite à développer d'autres opérateurs permettant de répondre convenablement à toutes ces exigences.



## CHAPITRE III

### DETECTION DES CONTOURS PAR OPERATEURS MORPHOLOGIQUES EN VISION LINEAIRE.

#### I. INTRODUCTION.

Dans le chapitre précédent, nous avons constaté que les techniques classiques de segmentation des images acquises par caméra linéaire atteignent leur limite d'efficacité dans les cas où la composante "éclairage" du signal vidéo initial présente de nombreuses variations spatiales et temporelles.

La Morphologie Mathématique est une approche originale du traitement de l'image qui tend à résoudre ces problèmes. En effet, les opérations de la Morphologie Mathématique permettent de réduire de façon notable la quantité importante d'informations contenues dans une image, en éliminant les informations parasites tout en préservant les caractéristiques des formes présentes [HAR-87]. Ceci apparait comme une approche plus naturelle du traitement de l'image.

Cette approche n'est pas nouvelle. En effet, parmi les premiers opérateurs appliqués sur les images numériques, nous trouvons les opérateurs morphologiques [KIR-57], [UNG-58], [MOO-68]. Les applications des opérateurs morphologiques sont très nombreuses : dans les domaines du biomédical [STE-83], [ROD-83], [MEY-79], [SKO-86], de l'inspection automatique [STE-83], [MAN-85], de la reconnaissance de formes [CRI-85], du filtrage non-linéaire des signaux [MAR-85], [MAR-87],

du traitement du bruit [GOE-85], [MAR-86], [ROS-82], de l'analyse de texture [WER-85], des formes floues [SER-82], [MAR-86], [SUZ-85]...

Le champ d'application de la Morphologie Mathématique appliquée au traitement des images s'est considérablement développé grâce aux progrès technologiques accomplis dans les architectures des systèmes informatiques. En effet, les temps de traitement importants des opérations morphologiques ont poussé les chercheurs et les industriels à développer des processeurs spécifiques et des architectures informatiques très élaborées afin d'en réduire les temps de calculs et par conséquent, augmenter les performances des opérateurs morphologiques [GOL-69], [GRA-80], [KRU-77], [KLE-77], [DUF-79], [GER-81], [DAN-81], [KIM-85], [MCC-85], [WIL-85], [LEO-85], [PRA-85], [HAR-85], [GER-84].

Ces opérateurs sont développés à partir d'une théorie bien définie de la Morphologie Mathématique. Celle-ci s'inspire directement de la Théorie des ensembles. De nombreux ouvrages ont été édités sur cette théorie mais peu d'entre eux ne la présentent en corrélation avec le traitement des images numériques. Parmi les ouvrages établissant le lien avec les images numériques, notons ceux de Serra [SER-82] et Matheron [MAT-75] qui, pour beaucoup, tiennent lieu de référence pour la Morphologie Mathématique appliquée au traitement des images numériques. Notons également, dans le domaine plus large du traitement du signal, les travaux de Maragos et Schafer [MAR-85], [MAR-86], [MAR-86].

Dans ce chapitre, nous n'avons pas la prétention de présenter tous les aspects de la Morphologie Mathématique. Nous nous contenterons d'introduire les définitions de base nécessaires au développement des opérateurs de détection des contours que nous proposons dans la suite de ce chapitre.

## II. NOTIONS PRELIMINAIRES.

Dans tout ce qui suit, nous travaillons sur des images numérisées sur une trame carrée.  $p$  indique un point élémentaire de coordonnées  $(x, y)$  dans l'image  $I$ .

Une des notions les plus importantes en Morphologie Mathématique est celle d'**élément structurant**. Il s'agit d'une configuration géométrique particulière de points à rechercher dans une image.

On appelle **centre de l'élément structurant**, un point privilégié de ce dernier qui est généralement le centre géométrique, mais qui peut tout aussi bien se trouver ailleurs et même sur sa périphérie pour mettre en valeur certaines propriétés topologiques. Un élément structurant centré en  $p$  sera noté  $B_p$ .

Dans le cas d'une image binaire, l'**élément structurant** est également défini par la disposition des points blancs qui le constituent, codés par des 1. (cf figure 5 )

```

.....
.. 1 1 1 ..
.. 1 1 1 ..
.. 1 1 1 ..
.....

```

```

.....
.. 1 1 1 ..
... 1 .....
.....
.....

```

1 : point élémentaire blanc de l'élément structurant.

1 : centre de l'élément structurant.

figure 5 : deux exemples d'éléments structurants.

Le symétrique de  $B_p$  par rapport à son centre est appelé le transposé de  $B_p$ .  
( cf figure 6 )

.....  
... 1 1 .  
.....

.....  
.. 1 1 ..  
.....

figure 6 : un élément structurant plan et son transposé.

### III. MORPHOLOGIE MATHÉMATIQUE POUR LES IMAGES BINAIRES.

Pour la clarté de l'exposé, nous présentons d'abord quelques aspects de la morphologie mathématique pour les images binaires avant d'aborder les images en niveaux de gris.

En Morphologie pour les images binaires, nous cherchons à retrouver la configuration d'un élément structurant défini dans une image  $I$  en fonction de l'opération à effectuer.

Sans perte de généralité, nous considérons une image binaire  $I$  constituée de deux types de régions distinctes : le fond et la forme. Sous le vocable "forme", on désigne en fait l'ensemble  $F$  de toutes les formes présentes dans la scène. Par convention, nous considérons une forme blanche sur fond noir. On affecte donc à l'ensemble des points constituant la forme la valeur 1 alors que les points du fond sont caractérisés par la valeur 0.

Les opérations de base consistent à examiner le voisinage de chaque point de l'image  $I$ , afin d'y reconnaître, éventuellement, la configuration

définie par l'élément structurant. On étiquette le centre de l'élément soit avec la valeur 0, soit avec la valeur 1 en fonction de l'opération souhaitée.

Selon la loi choisie, on aboutit à des opérations de dilatation ou d'érosion.

### III.1. La dilatation.

#### III.1.1. Origine.

La définition théorique fut proposée par H. Minkowski [MIN-03]. Dans le domaine du traitement de l'image, la dilatation fut utilisée très tôt par certains chercheurs comme opérateur de lissage de l'image [KIR-57], [UNG-57], [MOO-68], [GOL-69], [PRE-61], [PRE-73]. Nous la retrouvons en morphologie appliquée aux images binaires dans les travaux de Matheron [MAT-75] et Serra [SER-82].

#### III.1.2. Principe.

Quand l'élément structurant  $B_p$ , de centre  $p$  est reconnu dans le voisinage du point  $p$  de l'image  $I$ , on assigne à ce point la valeur 1. Par contre, si l'élément structurant n'est pas reconnu dans le voisinage du point  $p$ , on se pose la question suivante :

$B_p$  a-t-il des points appartenant à la forme ?

ou en d'autres termes :

$B_p$  touche-t-il la forme ?

Si la réponse est positive, on assigne la valeur 1 au point  $p$ , sinon la valeur 0.

Par cette opération, nous obtenons un ensemble, noté  $D$ , qui est l'ensemble "dilaté" de  $F$  par l'élément structurant  $B_p$  et qui se formule comme suit :

$$D = \{p: B_p \cap F \neq \emptyset\}$$

## III.2. L'érosion.

### III.2.1. Origine.

Le première définition de l'érosion fut proposée par R.M. Hadwiger [HAD-57]. Mais la définition qui s'applique directement au traitement des images numériques est celle proposée par Matheron [MAT-75].

En Morphologie appliquée aux images binaires, l'érosion est l'opération duale de la dilatation. Ainsi, l'opération d'érosion n'est pas indépendante de l'opération de dilatation. En effet, si nous érodons l'ensemble complémentaire de  $F$  par l'élément structurant  $B_p$ , nous obtenons le même résultat que si nous dilations  $F$  par le transposé de  $B_p$ .

Ces deux opérations morphologiques en images binaires, l'érosion et la dilatation, sont donc duales par rapport à la complémentarité.

### III.2.2. Principe.

L'opération d'érosion se définit de manière analogue. Si l'élément structurant  $B_p$ , de centre  $p$  est reconnu dans le voisinage du point  $p$ , on se pose la question :

$B_p$  est-il strictement inclus dans la forme ?

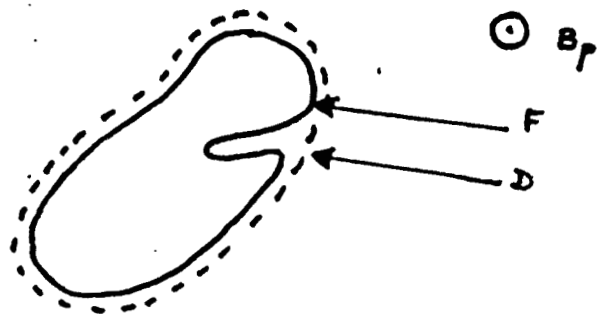


figure 7 : dilatation de la forme  $F$  par l'élément structurant  $B_p$ .

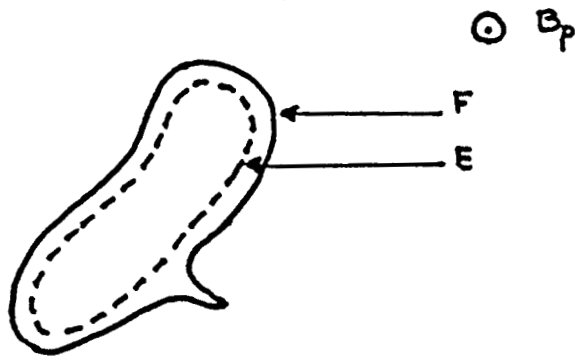


figure 8 : érosion de la forme  $F$  par l'élément structurant  $B_p$ .

Si la réponse est positive, on assigne la valeur 1 au point  $p$ , et dans le cas contraire, la valeur 0. L'ensemble  $F$  est ainsi "érode". L'érodé de  $F$  par l'élément structurant  $B_p$ , noté  $E$ , peut être caractérisé de la façon suivante:

$$E = \{ p : B_p \subset F \}$$

#### IV. MORPHOLOGIE MATHÉMATIQUE POUR LES IMAGES EN NIVEAUX DE GRIS.

Les opérateurs de la Morphologie Mathématique pour les images binaires agissent sur des ensembles de points élémentaires, auxquels sont associées les valeurs 1 ou 0, selon qu'ils appartiennent à la forme ou au fond. Certains concepts de la Morphologie Mathématique binaire peuvent être étendus aux images en niveaux de gris. Ainsi, apparues les opérations de dilation et d'érosion étendues aux images en niveaux de gris appelées ici opérations Max et Min [NAK-78]. Cette extension fut par la suite largement développée [STE-80], [STE-82], [PEL-81], [WER-85].

Comme pour les images binaires, le principal apport de la Morphologie Mathématique se situe au niveau de l'utilisation de l'élément structurant qui possède, comme précédemment, un centre qui n'est pas forcément son centre géométrique. Les opérateurs associés à cet élément structurant permettent d'éroder ou de dilater l'image en niveaux de gris.

##### IV.1. La dilatation.

Pour dilater une image en niveaux de gris, on considère un élément structurant centré en chacun de ses points. Dans le domaine délimité par cet élément structurant  $B_p$  centré au point  $p$ , la fonction de niveau de gris présente une valeur maximale.

En attribuant cette valeur maximale du niveau de gris au centre de l'élément structurant, on obtient la fonction dilatée, notée  $D[f(p)]$ . Cette opération peut se formuler de la manière suivante :



$$D [f(p)] = \sup \{ f(u) : u \text{ appartenant à } B_p \}$$

Par cette opération, la surface des régions sombres diminue tandis que celle des régions claires augmente. De plus, cet opérateur permet d'éliminer toutes les régions de surface inférieure à celle de l'élément structurant. Cette propriété est importante pour éliminer les points isolés de l'image.

#### IV.2. L'érosion.

De manière analogue, la fonction de niveau de gris prend, dans le domaine défini par l'élément structurant, une valeur minimale. Pour construire la fonction de niveau de gris érodée, on attribue cette valeur minimale au centre de l'élément structurant.

En notant  $E [f(p)]$  cette fonction érodée, il vient :

$$E [f(p)] = \inf \{ f(u) : u \text{ appartenant à } B_p \}$$

Par cette opération, la surface des régions claires diminue tandis que celle des régions sombres augmente. De plus, cet opérateur permet d'éliminer toutes les régions de taille inférieure à celle de l'élément structurant. Cette propriété est très utile pour éliminer certaines composantes du bruit, de type "salt and peper", présentes dans une image.

#### IV.3. Conclusion.

On constate que ces deux opérateurs morphologiques en niveau de gris présentent des complémentarités très intéressantes. Chacun des deux permet d'éliminer des composantes du bruit et d'améliorer l'homogénéité des régions présentes dans l'image.

Cependant, ce ne sont pas ces propriétés qui nous ont amenés à utiliser ce type d'opérateur pour détecter les contours des images en niveaux de gris. La propriété que nous exploitons réside dans le phénomène d'érosion

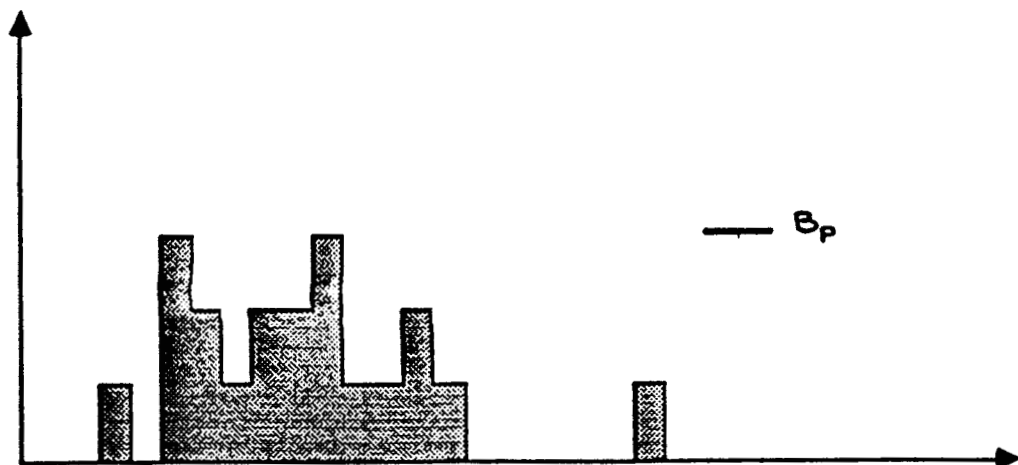


figure 9 : une fonction de niveau de gris.

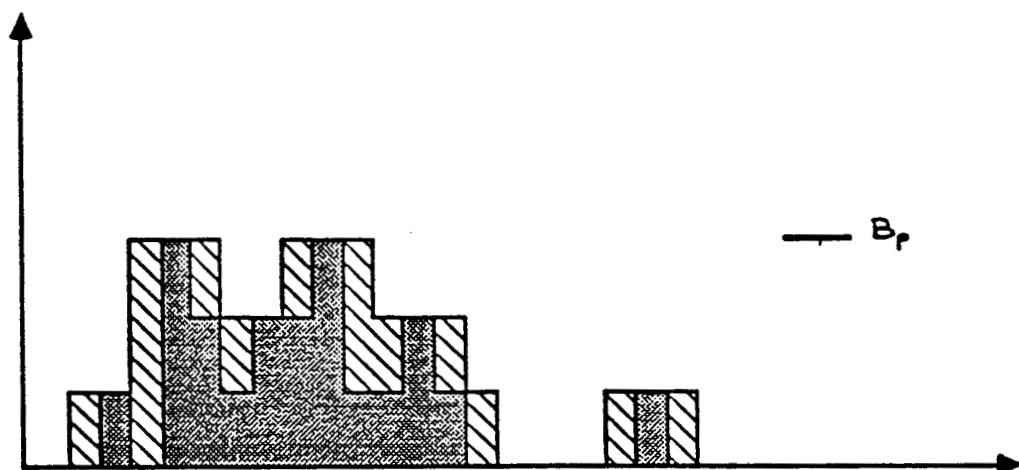


figure 10 : dilatation d'une fonction de niveau de gris par l'élément structurant plan  $B_p$ .

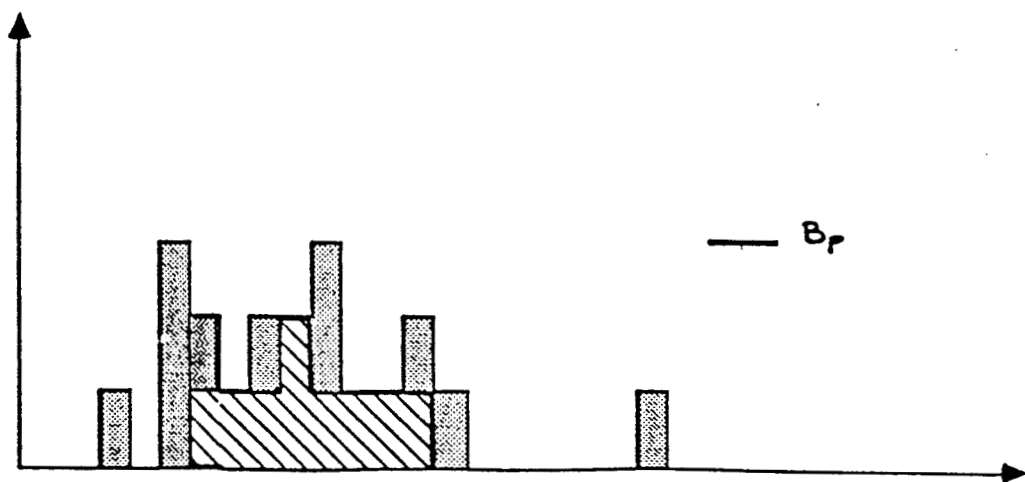


figure 11 : érosion d'une fonction de niveau gris par l'élément structurant

et de dilatation des zones claires et sombres dans les régions à fortes variations locales de niveaux de gris.

En effet, dans ces régions, le résultat de l'érosion a tendance à abaisser fortement le niveau de gris alors que la dilatation a tendance à le renforcer de façon notable.

Ce phénomène peut être exploité pour obtenir un opérateur sensible à la présence des contours dans l'image.

## V. OPERATEUR MORPHOLOGIQUE DE DETECTION DES CONTOURS.

### V.1. Principe.

Nous avons montré que les opérations d'érosion et de dilatation affectent surtout la périphérie des régions homogènes de l'image, c'est à dire, là où de fortes variations locales du niveau de gris indiquent la présence de contours. De plus, ces opérateurs ne modifient que légèrement l'intérieur des régions. En effet, ce n'est que dans le voisinage du contour que la valeur maximale et la valeur minimale de niveau de gris dans l'élément structurant peuvent être sensiblement différentes de la valeur originale du niveau de gris au centre de cet élément.

Tirant parti de cette remarque, on peut quantifier l'homogénéité du niveau de gris dans le voisinage du point  $p$  défini par l'élément structurant  $B_p$  par :

$$H_e = [ f(p) - E [f(p)] ]$$

dans le cas de l'opération d'érosion.  $H_e$  représente l'écart entre le niveau de gris du point  $p$  et le niveau de gris minimal dans le domaine délimité par l'élément structurant  $B_p$  centré au point  $p$ .

De même, dans le cas d'une dilatation, nous obtenons :

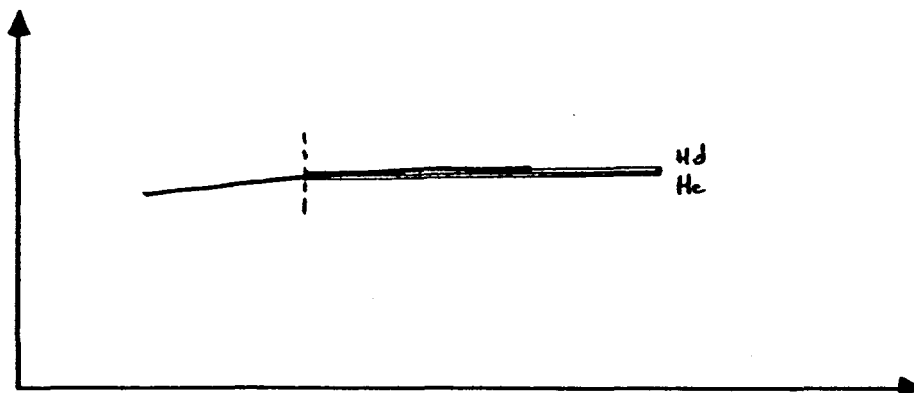
$$H_d = [ D [f(p)] - f(p) ]$$

$H_d$  représente l'écart entre le niveau de gris maximal dans le domaine délimité par l'élément structurant  $B_p$  centré au point  $p$  et le niveau de gris de ce point.

Ces deux écarts,  $H_e$  et  $H_d$ , permettent de localiser les contours des régions de l'image. En déplaçant l'élément structurant, on dispose des valeurs de  $H_e(p)$  et de  $H_d(p)$  en tous les points de l'image. On peut se trouver en présence de 4 types de situations :

**Situation de type 1 :**

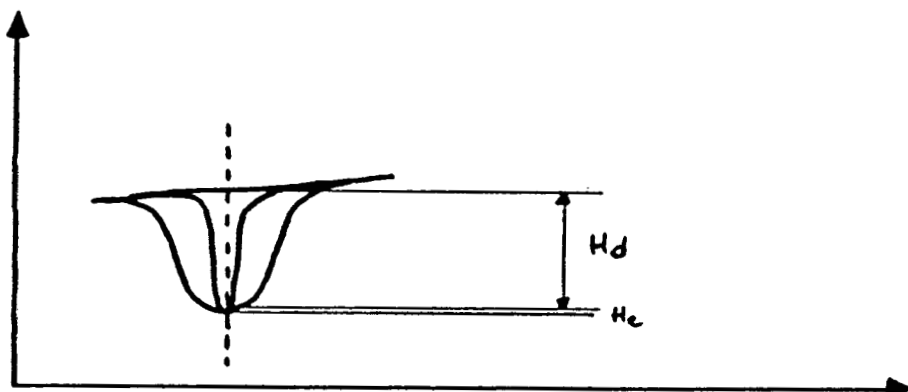
$H_e(p)$  et  $H_d(p)$  présentent des valeurs faibles.



Dans ce cas, l'élément structurant est inclus dans une région homogène en niveau de gris.

**Situation de type 2 :**

$H_d(p)$  présente une valeur élevée par rapport à celle de  $H_e(p)$ .

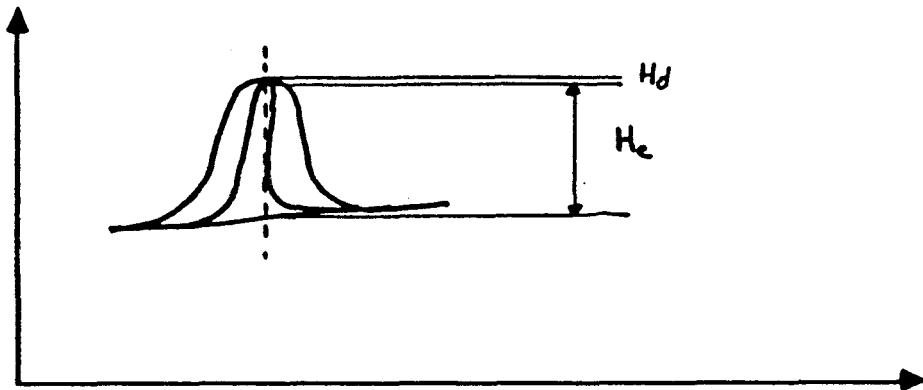


Ce cas correspond à un point sombre dans un environnement de points clairs.  $H_d(p)$  représente l'écart de niveau de gris qui existe entre le point sombre  $p$  et le point le plus clair dans le domaine délimité par l'élément structurant.

Cette configuration correspond à quelques points sombres isolés dans une région claire qui peuvent être assimilés à du bruit. L'opérateur de détection que nous cherchons à mettre en oeuvre ne doit pas réagir dans ce cas malgré la présence d'une forte variation locale de niveau de gris.

**Situation de type 3 :**

$H_e(p)$  présente une valeur élevée par rapport à celle de  $H_d(p)$ .

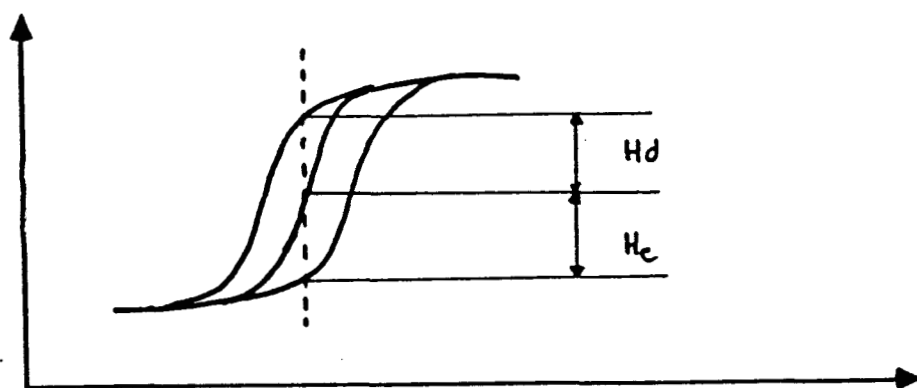


Ce cas correspond à un point clair dans un environnement de points sombres.  $H_e(p)$  représente l'écart de niveau de gris qui existe entre le point clair  $p$  et le point le plus sombre dans le domaine délimité par l'élément structurant.

Cette configuration correspond à quelques points clairs isolés dans une région sombre qui peuvent être considérés comme du bruit. L'opérateur de détection ne doit pas non plus réagir dans ce cas.

**Situation de type 4 :**

Cette dernière situation correspond au cas où  $H_e(p)$  et  $H_d(p)$  prennent simultanément des valeurs élevées.



Ce cas correspond à la présence effective d'un contour, c'est à dire d'une discontinuité brutale entre 2 régions distinctes de niveaux de gris homogènes.

C'est seulement dans ce cas que l'opérateur que nous désirons construire doit répondre afin de révéler les points de contour de l'image.

L'analyse des 4 types de situations rencontrées dans des images bruitées nous conduit à utiliser l'opérateur suivant, noté DOE  $[f(p)]$ :

$$\text{DOE } [f(p)] = \text{MIN} \{ [f(p) - \inf(f(u))], [\sup(f(u)) - f(p)], u \text{ appartenant à } B_p \}$$

pour détecter les contours.



DOE [f(p)] est donc la valeur minimale entre :

- l'écart entre le niveau de gris au point p et le niveau de gris qui lui est assigné par l'opération d'érosion par l'élément structurant centré en ce point et,

- l'écart entre le niveau de gris assigné au point p par l'opération de dilatation par le même élément structurant et le niveau de gris initial de ce point.

Cet opérateur ne prend des valeurs élevées que pour les points de contour effectifs et ne réagit pas à la présence de points isolés dûs au bruit.

Cette propriété constitue un avantage déterminant sur les opérateurs différentiels classiques qui répondent de manière systématique à toute variation locale de niveau de gris, surtout lorsque la fenêtre de calcul est de taille réduite.

Les résultats présentés au paragraphe suivant illustrent le comportement de cet opérateur morphologique de détection des contours.

## VI. EVALUATION DE L'OPERATEUR.

L'évaluation quantitative des performances des opérateurs de détection des contours peut être effectuée sur une base déterministe en mesurant l'amplitude et l'orientation de la réponse des opérateurs sur des images types [ADB-79]. La comparaison statistique des détections correctes et des fausses détections constitue une autre voie d'évaluation [ADB-79], [FRA-75], [DEU-78]. Cependant une étude critique [ADB-79] de ces techniques d'évaluation montre qu'elles ne permettent pas de différencier de bons opérateurs tels que ceux de Sobel [DUD-73] et de Prewitt [PRE-70].

C'est la raison pour laquelle nous proposons un autre type d'évaluation, en seillant les réponses de différents opérateurs de telle sorte que le nombre de points de contour retenus soit le même pour chacun d'entre eux. L'examen visuel de l'image ainsi obtenue permettra alors de

comparer les performances de différentes techniques de détection de contour.

### **VI.1. Détection de contour dans une fenêtre 3x3.**

Il ressort de l'étude d'Abdou et Pratt [ADB-79] que les opérateurs de Sobel et Prewitt sont les plus performants dans la catégorie des opérateurs différentiels dans une fenêtre 3 x 3.

C'est donc à ces opérateurs, souvent cités comme référence, que nous comparons l'opérateur DOE proposé.

Pour ce faire, nous utilisons les images de la figure 12 qui ont été acquises sur du tissu imprimé en défilement continu sur une table de découpe laser.

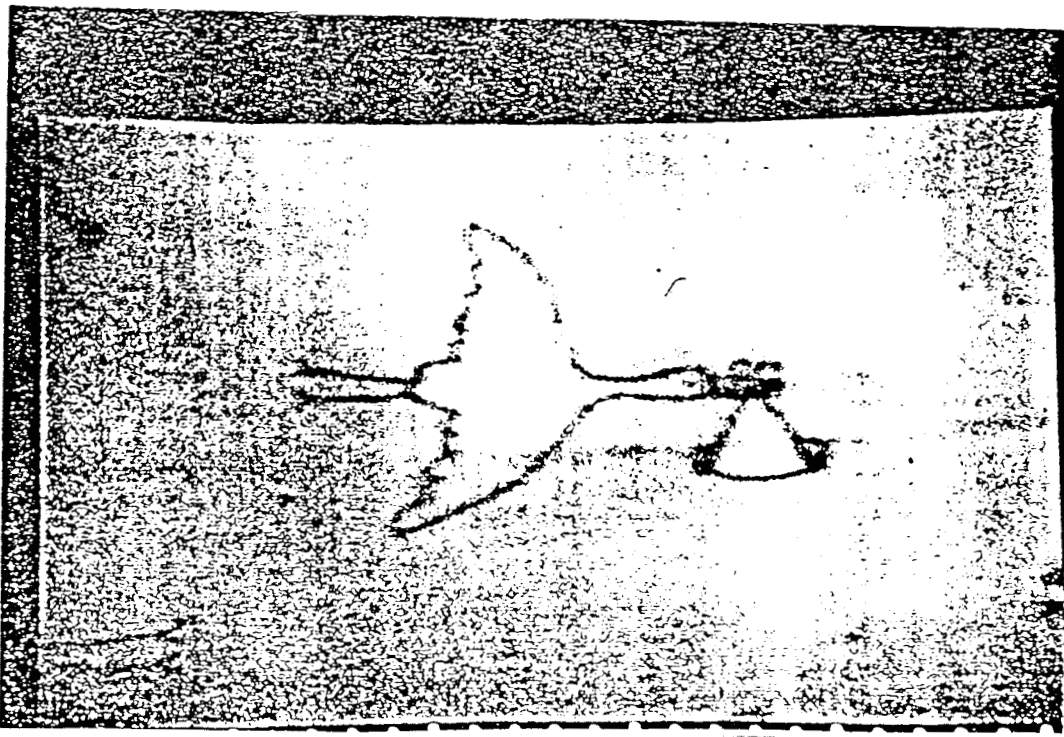
Les 3 opérateurs de détection de contour ont été appliqués à l'image 12.a. Les réponses de ces opérateurs, seuillées à différents niveaux de telle sorte que le nombre de points de contour retenus soit égal à 3000, sont présentées sur les figures 13, 14 et 15.

On constate que l'opérateur morphologique permet de détecter le dessin intérieur de la voile du navire ( cf figure 13 ) alors que ce dessin n'apparaît pas avec l'emploi des deux autres opérateurs ( cf figures 14 et 15 ). On constate également que le mat du bateau, le carré oblique le plus à droite de l'image et le bord intérieur du carré de gauche sont mieux restitués sur la figure 13 que sur les deux autres.

Si l'on désire faire apparaître le motif interne de la voile et améliorer la connexité du mat, du carré oblique et du bord intérieur du carré de gauche, on peut abaisser le seuil afin d'augmenter le nombre de points de contour retenus. Les images des figures 16 et 17 montrent ces résultats pour les opérateurs de Sobel et de Prewitt avec 4000 points au-dessus du seuil. La connexité est effectivement meilleure mais ce résultat a été obtenu au détriment de la finesse des contours.



figures 12 a et b : images de tissu imprimé en défilement continu.



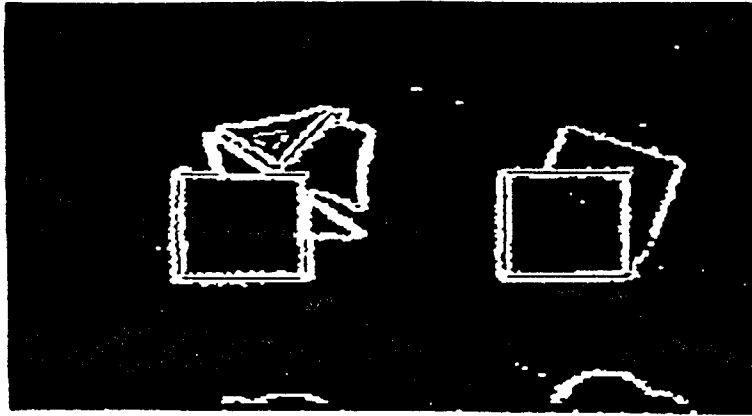


figure 13 : opérateur morphologique appliqué sur la fig 12.a. fenêtre 3x3.

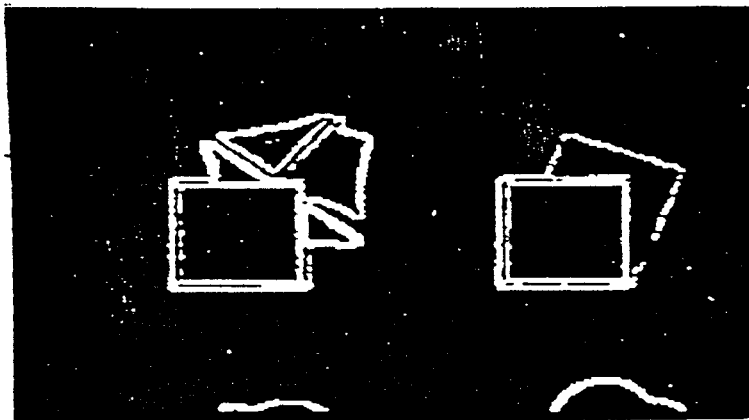


figure 14 : opérateur de Sobel appliqué sur la fig 12.a. fenêtre 3x3.

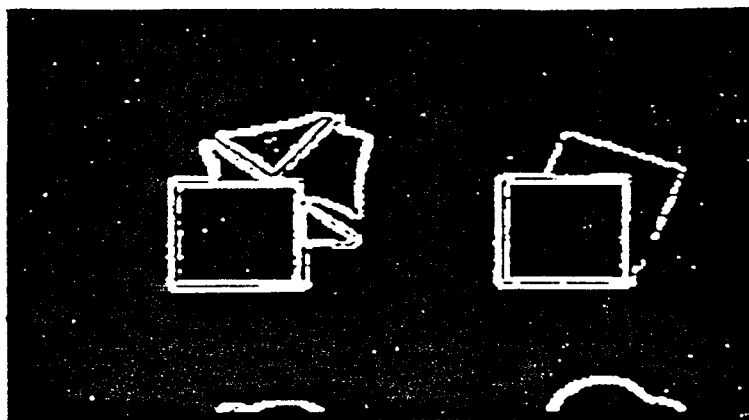


figure 15 : opérateur de Prewitt appliqué sur la fig 12.a. fenêtre 3x3.

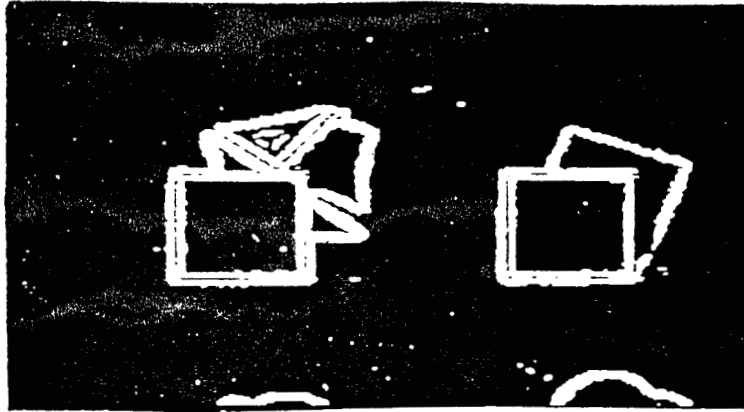


figure 16 : opérateur de Sobel appliqué sur la fig 12.a. fenêtre 3x3.

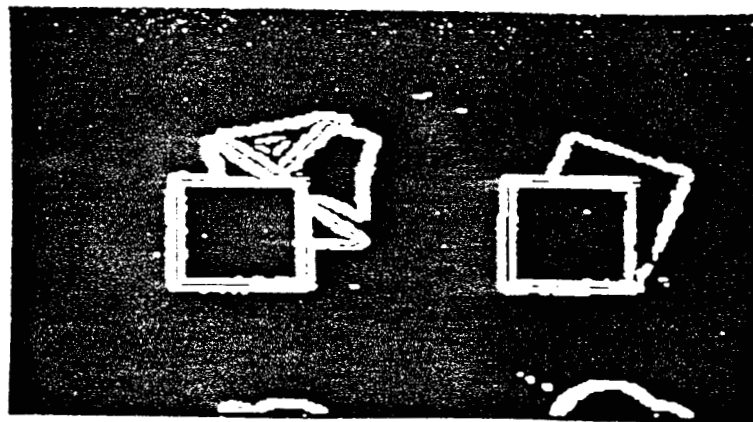


figure 17 : opérateur de Prewitt appliqué sur la fig 12.a. fenêtre 3x3.

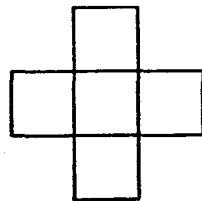
Les mêmes traitements ont été effectués sur l'image b de la figure 12 ( la cigogne ). Avec 1000 points au dessus du seuil de binarisation ( cf figures 18, 19 et 20 ), on constate la supériorité de l'opérateur morphologique sur ceux de Sobel et Prewitt.

Comme précédemment, en abaissant le niveau du seuil, on améliore la connexité des contours obtenus par les opérateurs de Sobel et Prewitt au détriment de la finesse des contours ( cf figures 21 et 22 )

## VI.2. Détection des contours sur une fenêtre cruciforme.

Dans l'optique d'une implantation temps réel de l'opérateur DOE, nous utilisons une fenêtre plus réduite qui limitera les temps d'accès aux points voisins du point courant.

Afin de conserver un opérateur réellement bidimensionnel, nous utilisons la fenêtre cruciforme représentée ci-dessous :



Avec cette fenêtre, le seul opérateur différentiel utilisable est défini par les masques classiques :

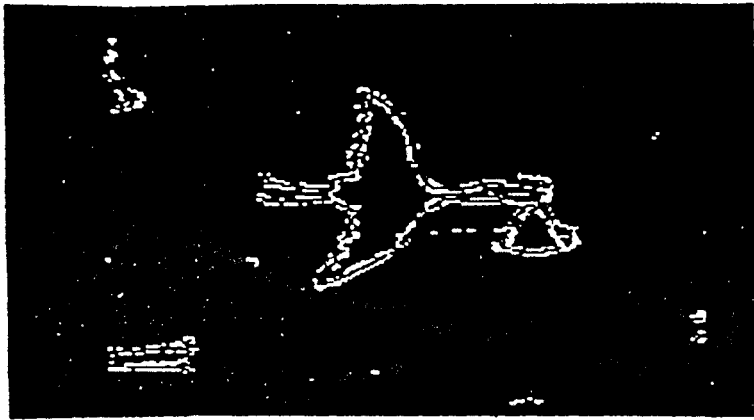


figure 18 : opérateur morphologique appliqué sur la fig 12.b. fenêtre 3x3.

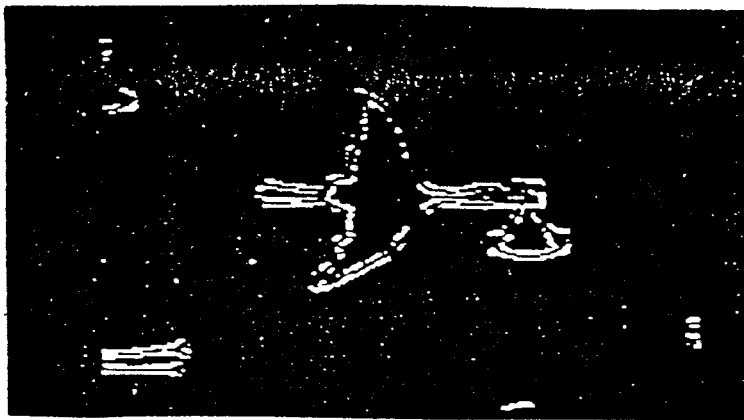


figure 19 : opérateur de Sobel appliqué sur la fig 12.b. fenêtre 3x3.

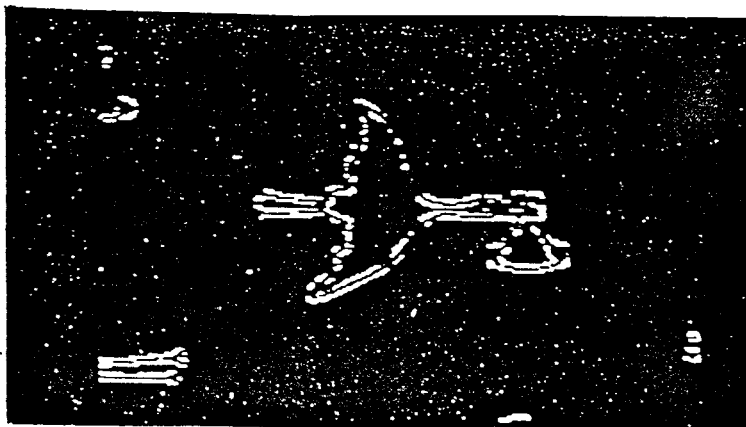


figure 20 : opérateur de Prewitt appliqué sur la fig 12.b. fenêtre 3x3.

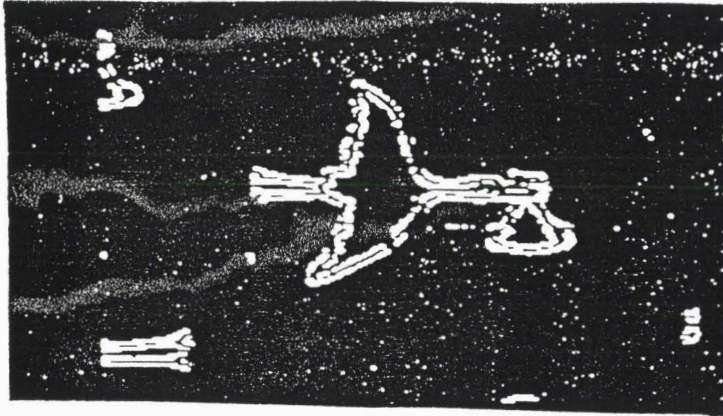


figure 21 : opérateur de Sobel appliqué sur la fig 12.b. fenêtre 3x3.

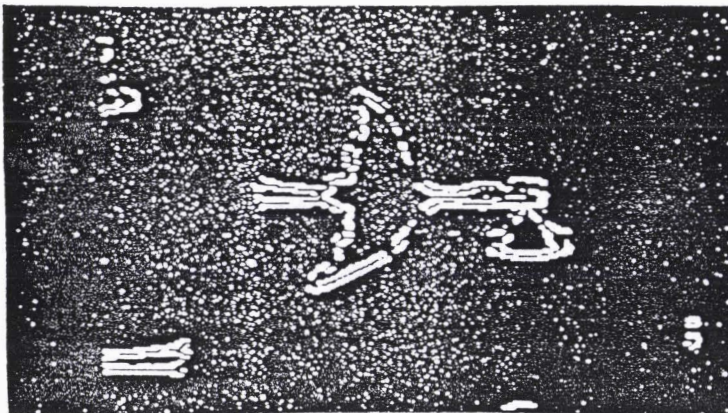
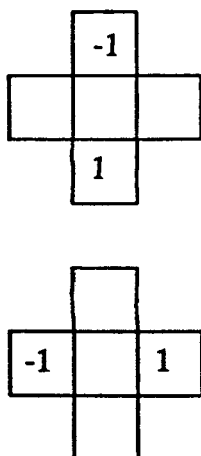


figure 22 : opérateur de Prewitt appliqué sur la fig 12.b. fenêtre 3x3.





Comme précédemment, nous allons comparer l'opérateur de détection morphologique de contour DOE à cet opérateur différentiel en utilisant les images de la figure 12.

Les figures 23 et 24 représentent les réponses de ces deux opérateurs seuillées de telle sorte qu'il ne subsiste que 2500 points de contour. Il est évident que l'opérateur morphologique est nettement supérieur à l'opérateur différentiel.

Les figures 25 a, b, c et d montrent qu'en augmentant le nombre de points de contour retenus avec l'opérateur différentiel, on épaissit les contours sans améliorer leur connexité. On fait alors apparaître un nombre important de points isolés dûs au bruit de l'image ( cf figures 26 et 27 ).

Pour l'image de la cigogne, on retient également la même supériorité de l'opérateur morphologique sur l'opérateur différentiel ( cf figure 9 (a) et (b)).

## VII. CONCLUSION.

L'opérateur morphologique de détection des contours présenté peut être aisément implanté dans un système de vision à caméra linéaire. Nous aborderons cette mise en oeuvre au chapitre VI.

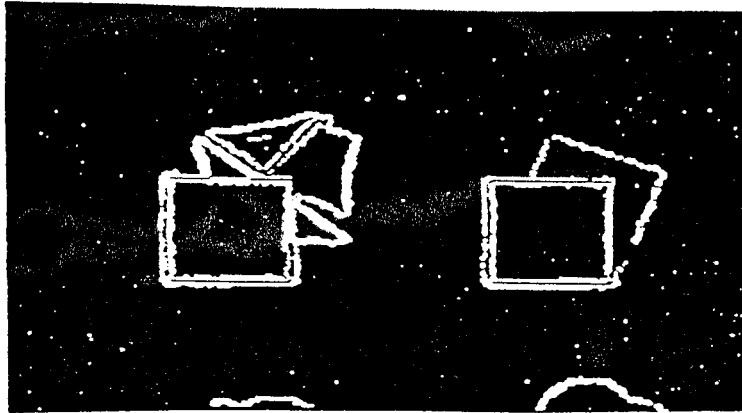


figure 23 : opérateur morphologique appliqué sur la fig 12.a. fenêtre cruciforme.

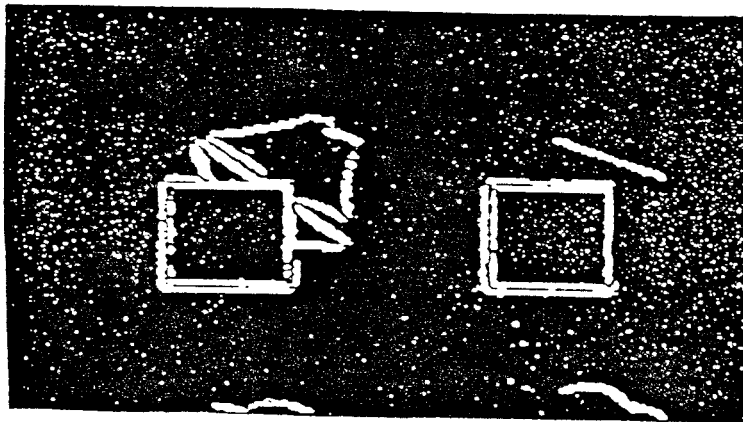
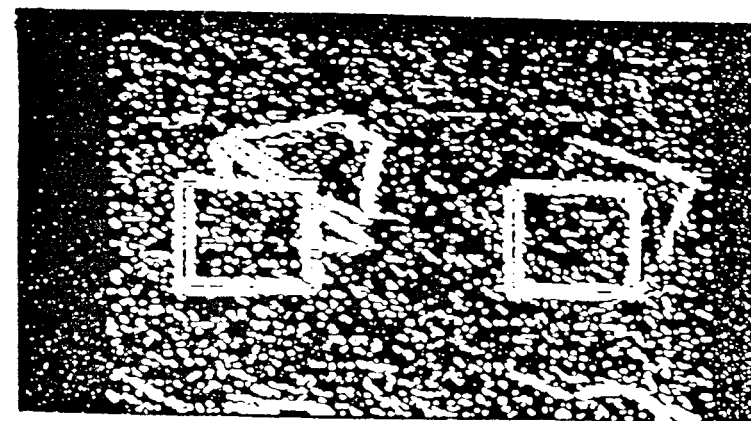
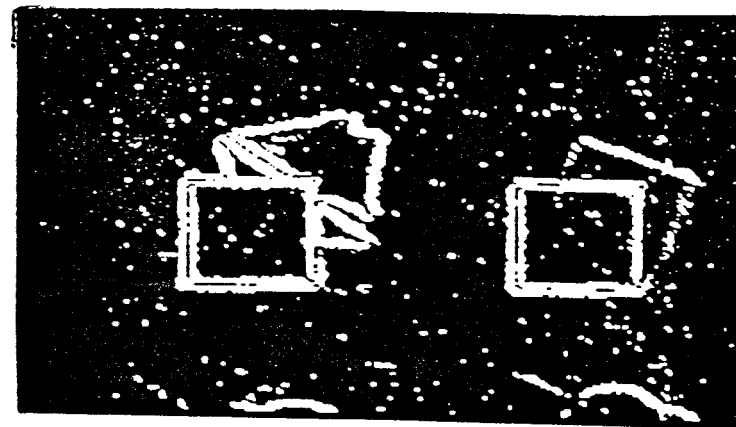
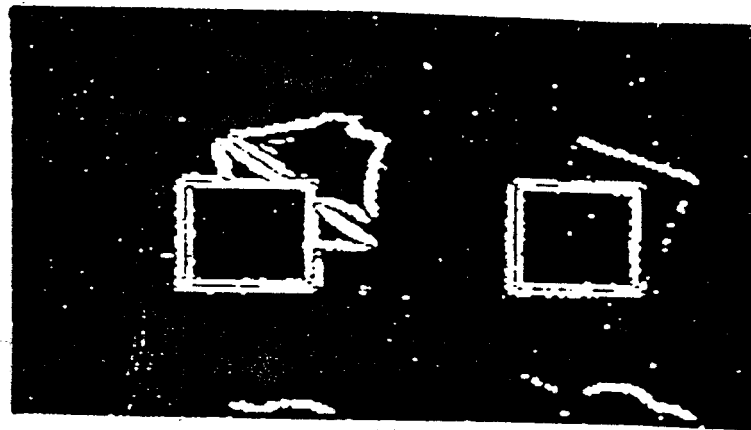
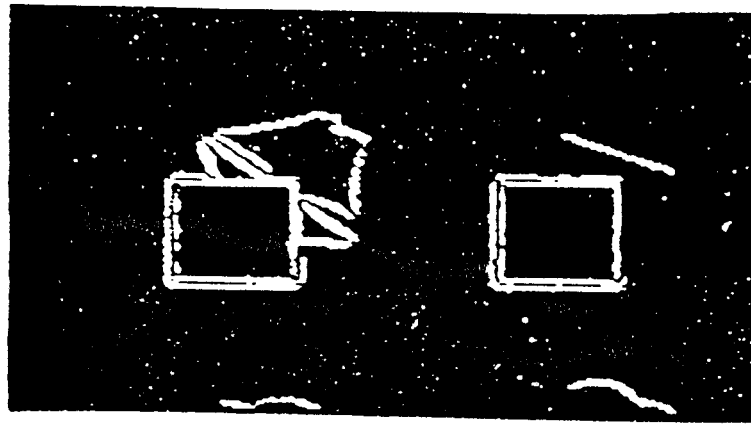


figure 24 : opérateur différentiel appliqué sur la fig 12.a. fenêtre cruciforme.



figures 25 a, b, c, d : opérateur différentiel appliqué sur la fig 12.a. fenêtre cruciforme.

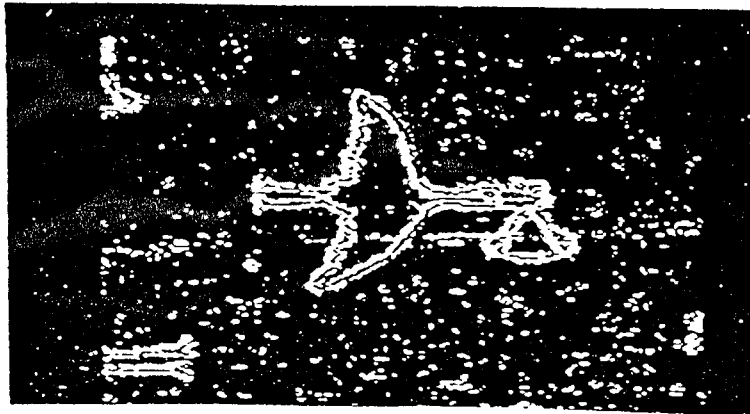


figure 26 : opérateur morphologique appliqué sur la fig 12.b. fenêtre cruciforme.



figure 27 : opérateur différentiel appliqué sur la fig 12.b. fenêtre cruciforme.

A l'issue du traitement de chaque image-ligne acquise par le système, on obtient une ligne dont les niveaux de gris représentent la réponse de l'opérateur proposé.

En seuillant cette réponse, il est aisé de détecter les points de contour présents dans la ligne. On obtient l'image-ligne des contours sous la forme d'une image-ligne binarisée.

Dans l'optique d'une reconstruction des contours en temps réel sur un système de vision linéaire, il s'agit maintenant de chaîner ces points de contour, de ligne en ligne, et de manière récurrente, afin d'obtenir les contours des formes en défilement sous la caméra.

## **CHAPITRE IV**

### **CHAINAGE DYNAMIQUE POUR LA RECONSTRUCTION DES CONTOURS.**

Dans le chapitre précédent, nous avons présenté une méthode permettant de détecter les points de contour présents dans une scène. Lorsque cette méthode est implantée sur un système de vision linéaire, on obtient des image-lignes où seuls les points de contour sont étiquetés. De telles image-lignes seront appelées **lignes de points de contour**.

Il s'agit maintenant de reconstituer les formes ou les objets à partir de leurs points de contour, de ligne en ligne, de manière récurrente, tout en respectant les contraintes du temps réel.

#### **I. METHODES USUELLES D'EXTRACTION DE FORME.**

Ce problème s'apparente au problème classique d'extraction de formes tel qu'on le rencontre en analyse d'images matricielles. En effet, il existe de nombreuses approches qui permettent d'extraire une forme complète d'une image à partir de la seule connaissance de ses points de contour.

Les techniques de remplissage ou de coloriage de formes délimitées par leurs contours, en synthèse d'image, utilisent souvent le balayage vidéo pour identifier les points de contour qui appartiennent à une même forme. La connaissance à priori d'un point intérieur à chaque contour présent dans l'image facilite énormément cette reconstitution de chaque forme.

Cependant, une telle procédure ne peut être envisagée que dans le cadre de systèmes de synthèse d'images à haut degré d'interactivité avec l'opérateur.

Plus proches de nos préoccupations, les techniques par tests de parité permettent de reconstituer des formes simples à partir de leurs contours. L'expérience montre cependant que ces techniques séquentielles, qui explorent le contour ligne par ligne en utilisant le balayage vidéo, ne permettent pas de traiter les contours de formes complexes. En effet, de nombreuses ambiguïtés dues à des arêtes horizontales, des sommets, des points doubles ou des recouvrements d'arêtes, etc... viennent facilement mettre ces méthodes en défaut [DUD-76], [ACK-81].

Notons que la décomposition en triangles des formes délimitées par leurs contours est la méthode la plus utilisée pour éliminer les difficultés rencontrées au niveau de l'identification des extrémités des segments intérieurs à une région [FUL-74], [ROM-69], [WYL-67], [CRO-77], [REY-75], [LIT-79], [NEW-73].

Les méthodes par inversion de l'image, qui consistent à explorer séquentiellement le contour et à inverser systématiquement tous les points à droite de celui considéré, ne s'appliquent que pour des configurations de contours simples [RAA-78], [MAR-82] ou demandent, dans des cas plus complexes, un prétraitement du contour [ACK-81].

Dans le contexte de la vision linéaire, l'utilisation de telles techniques nécessitent la mémorisation de nombreuses lignes de points de contour et leur traitement en temps différé, bloc par bloc. De plus, ces techniques sont très gourmandes en temps de calcul. Seules les méthodes par tri des intersections peuvent s'affranchir de cette contrainte, mais elles demeurent limitées aux contours polygonaux [REY-75], [NEW-73], [LIT-79].

Pour lever ces ambiguïtés sans émettre d'hypothèses restrictives, il est en général nécessaire de procéder en deux passes [PAV-78], [DIS-82]. Un étiquetage des points de contour au cours de la première phase en fonction de la configuration de leur voisinage permet ainsi de manipuler des formes extrêmement complexes [AME-86].

Cette double exploration des contours est tout à fait incompatible avec le principe même de la vision linéaire. Nous devons donc rechercher une méthode respectant toutes les exigences du temps réel et qui nous permette d'extraire au mieux les contours en vue d'une bonne caractérisation ultérieure des formes présentes dans la scène observée.

## II. CONFIGURATIONS DE BASE.

Afin de répondre aux contraintes imposées par l'utilisation d'un système de vision à caméra linéaire sans mémorisation de toute la scène observée, nous proposons une technique de reconstruction récurrente des contours développée initialement pour les images matricielles [BAJ-79], [BAJ-84], [CAT-85], [KIM-83], [RAK-79].

L'intérêt de cette technique est de ne nécessiter la mémorisation que de deux lignes de points de contour consécutives et de ne s'appuyer que sur des notions de connexité entre lignes.

Nous allons, dans un premier temps, examiner toutes les configurations possibles d'apparition des points de contour dans une fenêtre  $2 \times 2$  que nous déplacerons le long de deux lignes de points de contour consécutives. La dernière ligne acquise qui vient d'être traitée est appelée, dans la suite de l'exposé, **ligne courante**. Elle est notée **LC**. L'autre ligne, résultat du traitement de la ligne correspondant à l'acquisition précédant la ligne courante est appelée **ligne précédente** et elle est notée **LP**.

Par convention, un point appartenant au contour est étiqueté avec la valeur 1 et est schématisé comme suit :





Un point n'appartenant pas à un contour, étiqueté avec la valeur 0, est schématisé par :



Les configurations des points de contour des formes apparaissant dans la fenêtre d'observation 2 x 2 sont codées en utilisant une pondération qui permet de différencier les points élémentaires qui la constituent.

a	c
b	d

Cette pondération permet de générer un code qui sera utilisé pour caractériser les différentes configurations possibles.

Pour les différentes possibilités, le code retenu est donné par la formule :

$$Q = a \times 2^3 + b \times 2^2 + c \times 2 + d$$

où les paramètres a, b, c et d prennent les valeurs 1 ou 0 selon que les points correspondants appartiennent ou non au contour. Les 16 configurations possibles d'apparition de points de contour dans la fenêtre peuvent donc être caractérisées par 16 valeurs du code Q, de la valeur 0 à la valeur F en code hexadécimal. Le code Q peut donc être considéré comme un indicateur de l'état de la fenêtre.

Le déplacement de cette fenêtre d'observation le long des lignes de points de contour nous donne une succession d'états indiquant les

configurations de points de contour rencontrés. Il nous faut maintenant définir certaines notions importantes pour la suite du chapitre.

### II.1. Segment.

On appelle **segment** une série de points de contour adjacents sur une même ligne ( cf figure 28 ).

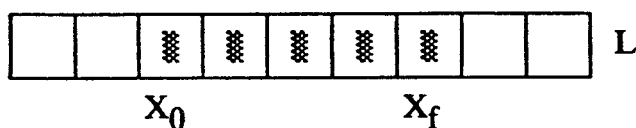


figure 28

Un segment est défini par l'ensemble des 3 paramètres :

$$(X_0, X_f, L)$$

avec

$X_0$  : le point d'origine du segment, c'est à dire le point le plus à gauche,

$X_f$  : le point final du segment, c'est à dire le point le plus à droite,

$L$  : le numéro de la ligne d'apparition du segment.

### II.2. Segments connexes.

Deux segments sur deux lignes consécutives sont dits **connexes** si au moins un point d'un de ces segments fait partie des 8 voisins d'un point de l'autre segment. La relation " le segment 1 est connexe au segment 2 " est réflexive, symétrique et transitive. Si le segment 1 sur une ligne est connexe au segment 2 sur une autre ligne, et si ce dernier est lui même connexe à un autre segment 3 de la première ligne, on dit que les segments 1 et 3 sont connexes. [POS-87]. ( cf figure 29 ).

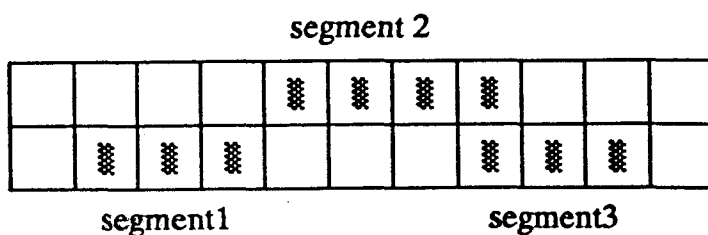


figure 29

Nous allons maintenant définir les configurations de base qui permettent de situer les différents segments dans le contour auquel ils appartiennent. Il importe de garder à l'esprit que les lignes de points de contour sont explorées séquentiellement, au fur et à mesure que la scène défile dans le champ de la caméra.

Le déplacement de la fenêtre d'observation le long de la ligne courante est repéré par l'abscisse  $X$  des éléments de poids les plus faibles de cette fenêtre, c'est à dire les éléments des positions  $c$  et  $d$  de la figure . La fenêtre correspondante est notée  $W(X)$ .

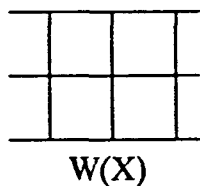
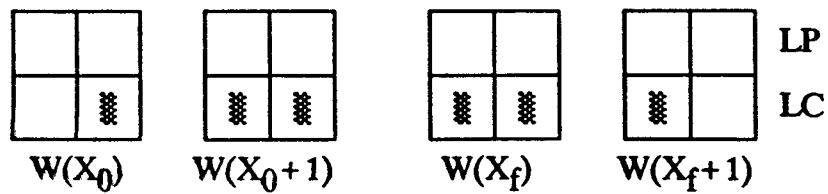


figure 30

### II.3. Début.

Nous appelons **début** l'apparition d'un segment sur la ligne courante LC qui n'a aucun segment connexe sur la ligne précédente LP. En déplaçant la fenêtre 2 x 2 le long des deux lignes consécutives, la séquence d'apparition des points qui caractérise un début est donc :



En respectant le codage préalablement défini, cette séquence est définie par la succession d'états suivants:

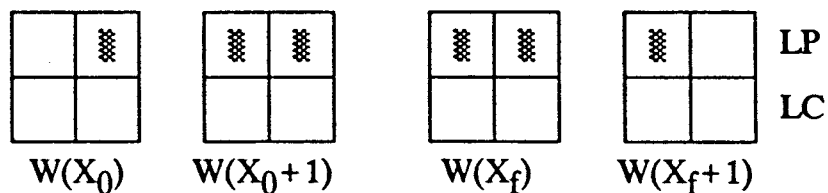
" 1, 5, ..., 5, 4 "

La fenêtre  $W(X_0)$  dont l'état est "1" indique l'abscisse du point origine  $X_0$  du segment "début" qui compte  $(k + 1)$  points, où  $k$  est le nombre de fenêtres correspondant à l'état " 5 ". Ce segment se termine au point d'abscisse  $X_{fin}$ . Il n'est donc identifié que lors de l'observation de la fenêtre  $W(X_f + 1)$ .

#### II.4. Fin.

Nous appelons **fin** la disparition d'un segment appartenant à la ligne précédente LP, c'est à dire un segment qui n'a aucun segment connexe sur la ligne courante LC.

La succession d'états relative à cette situation est donc :



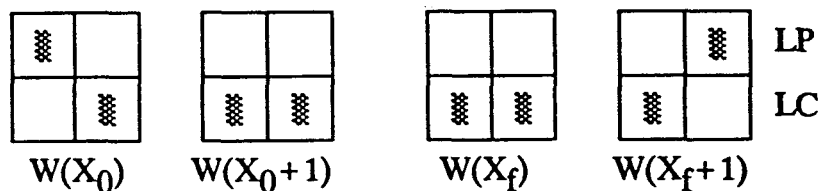
Son codage donne :

" 2, A, ..., A, 8 "

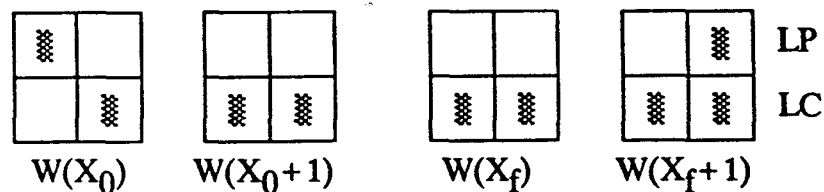
Ce segment "fin" compte  $(k + 1)$  points, où  $k$  est le nombre de fenêtres d'état "A". Le segment apparue à la ligne précédente LP et se terminant au point d'abscisse  $X_{fn}$ , n'est identifié qu'au niveau de la fenêtre  $W(X_f + 1)$  de la ligne courante LC.

### II.5. Réunion.

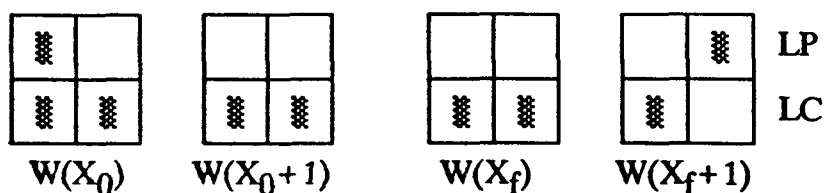
Lorsque 2 segments de la ligne précédente LP sont connexes à un troisième segment sur la ligne courante LC, nous avons vu que les 2 segments de la ligne précédente deviennent connexes. Ce résultat de la transitivité de la connexité est appelé réunion. La réunion se caractérise par plusieurs séquences et codages possibles.



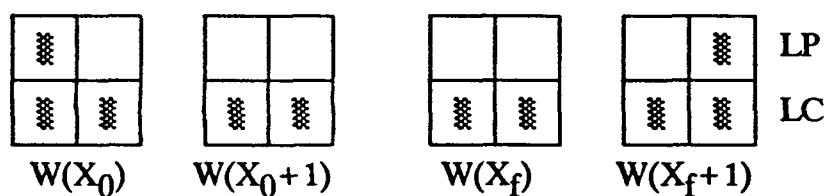
" 9, 5, ..., 5, 6 "



" 9, 5, ..., 5, 7 "



" D, 5, ..., 5, 6 "

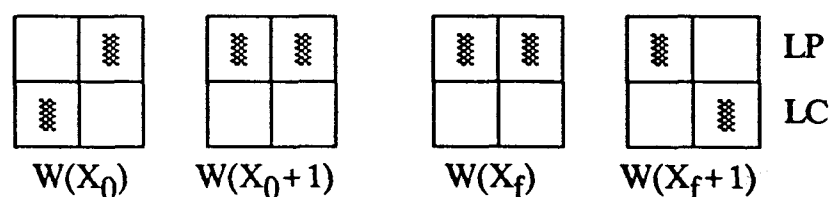


" D, 5, ..., 5, 7 "

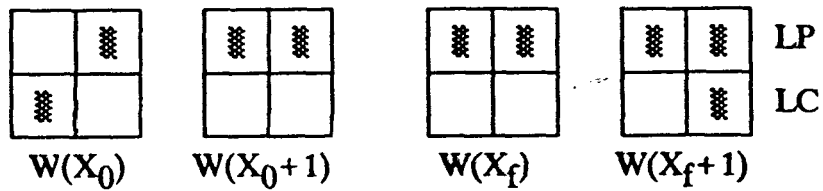
Le segment de la ligne précédente qui assure la réunion compte  $(k + 1)$  points, où  $k$  est le nombre de fenêtres d'état " 5 ".

## II.6. Séparation.

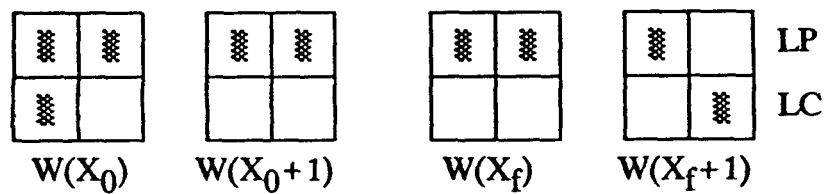
La séparation d'un segment présent à la ligne précédente LP en deux segments disjoints mais connexes à la ligne courante LC aboutit à plusieurs séquences et codages possibles.



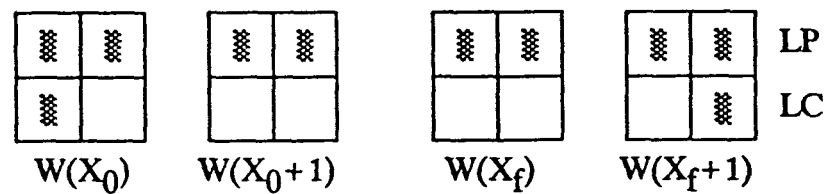
" 6, A, ..., A, 9 "



" 6, A, ..., A, B "



" E, A, ..., A, 9 "

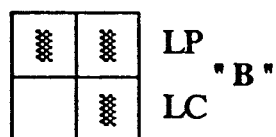
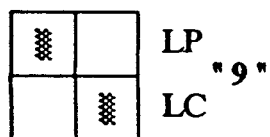
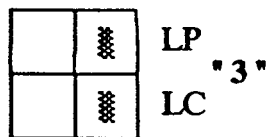
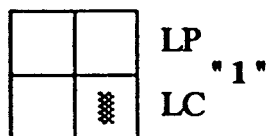


" E, A, ..., A, B "

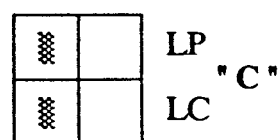
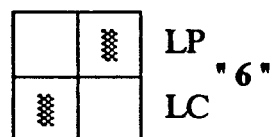
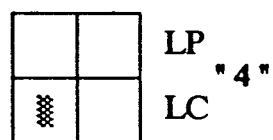
Le segment de la ligne précédente qui se scinde compte  $(k + 1)$  points, où  $k$  est le nombre de fenêtres d'état " A ".

## II.7. Entrée, Sortie.

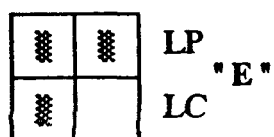
Enfin nous appelons respectivement **entrée** et **sortie**, le point origine et le point final sur la ligne courante LC. Les entrées sont caractérisées par les fenêtres d'observation et codages suivants :



Les sorties sont caractérisées par les fenêtres d'observation et codages suivants :

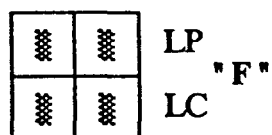
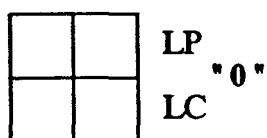






### II.8. Fenêtres vides et pleines.

Il ne faut pas oublier la fenêtre d'observation "vide" dans laquelle aucun point de contour n'apparaît. Son état est codé "0". L'état de la fenêtre "pleine", dans laquelle apparaissent quatre points de contour, est codé "F".



### II.9. Epaisseur.

Dans la suite, nous appelons épaisseur, une configuration qui est constituée d'une entrée suivie d'une série de fenêtres "pleines" et d'une sortie.

## III. EXPLORATION RECURRENTE DES LIGNES DE CONTOUR.

Le déplacement de la fenêtre d'observation 2 x 2 le long de deux lignes consécutives de points de contour et l'application des définitions

exposées ci-dessus nous permettent de chaîner les points de contour apparus sur la ligne courante LC avec ceux qui leur sont connexes sur la ligne précédente LP. Chaque segment est caractérisé soit comme étant un élément d'un contour déjà existant, soit comme étant le début d'un nouveau contour. Les lignes de points de contour provenant séquentiellement de la détection, les contours se retrouvent ainsi reconstitués, segment par segment, ligne après ligne.

Le codage de l'état de la fenêtre d'observation nous amène à construire un tableau indiquant toutes les transitions possibles entre les états successifs de la fenêtre au fur et à mesure de son déplacement le long des lignes précédente LP et courante LC. Dans le tableau ci-dessous, les états des colonnes sont les états possibles de la fenêtre  $W(X)$ . Les lignes indiquent les états possibles de la fenêtre suivante  $W(X+1)$ .

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	x				x				x				x			
1	x				x				x				x			
2	x				x				x				x			
3	x				x				x				x			
4		x				x				x					x	
5		x				x				x					x	
6		x				x				x					x	
7		x				x				x					x	
8			x				x				x					x
9			x				x				x					x
A			x				x				x					x
B			x				x				x					x
C				x				x				x				
D				x				x				x				
E				x				x				x				
F				x				x				x				

Ce tableau doit être exploité de la manière suivante : Supposons que la fenêtre  $W(X)$  soit dans l'état 0. La lecture de la colonne "0" du tableau indique qu'il n'y a que 4 transitions possibles vers les états 0, 1, 2 et 3 pour la fenêtre  $W(X+1)$ . Ces transitions peuvent être représentées sous la forme d'un Réseau de Petri ( cf figure 31 )

La place P0 indique que l'on a effectué la transition

$$W(X) = 0 \rightarrow W(X+1) = 0.$$

La place P1 indique que l'on a effectué la transition

$$W(X) = 0 \rightarrow W(X+1) = 1.$$

La place P2 indique que l'on a effectué la transition

$$W(X) = 0 \rightarrow W(X+1) = 2.$$

La place P3 indique que l'on a effectué la transition

$$W(X) = 0 \rightarrow W(X+1) = 3.$$

On note  $T_Q$  la transition qui amène l'état de la fenêtre  $W(X+1)$  à la valeur  $Q$ , quelque soit la place précédente de la fenêtre  $W(X)$ .

En appliquant cette technique de représentation à toutes les transitions possibles définies par le tableau, on aboutit finalement au Réseau de Pétri de la figure 32 .

Cette approche permet de ne pas stocker tous les états de la fenêtre rencontrés au cours de l'exploration des deux lignes. L'analyse de la séquence de ces états, qui ne nécessite que la mémorisation du dernier état, permet d'identifier les configurations de base définies précédemment.

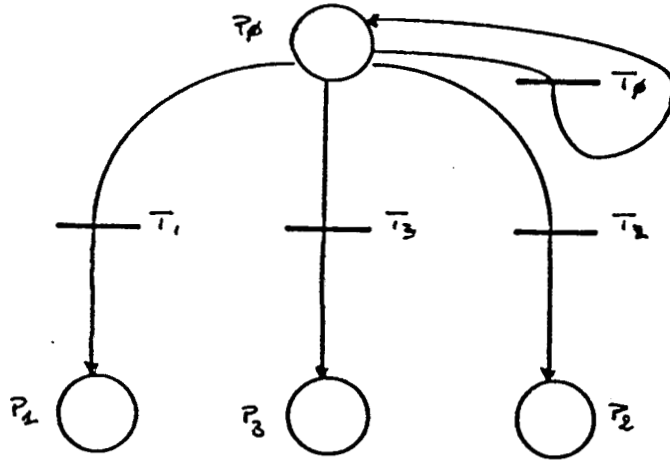


figure 31 : schématisation des transitions.

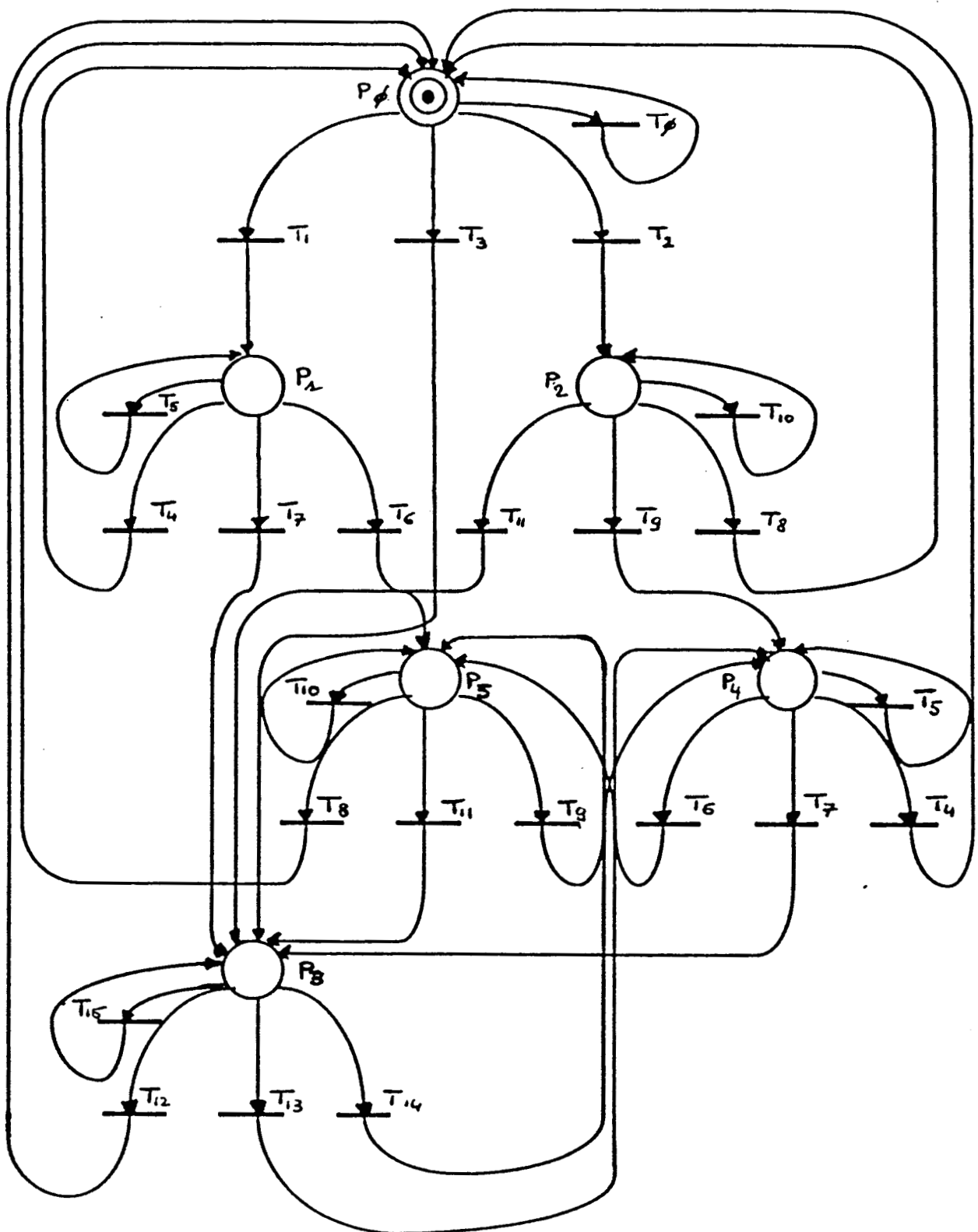
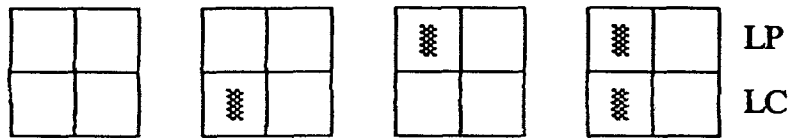
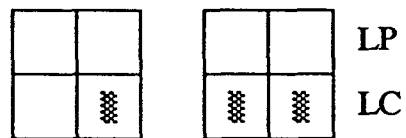


figure 32 : schématisation de l'exploration récurrente des lignes de points de contour.

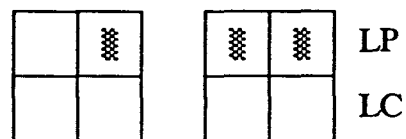
P0 : rupture de connexité.



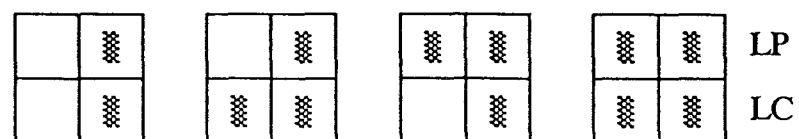
P1 : apparition d'un segment sur la ligne courante LC et poursuite de la connexité sur celle-ci.



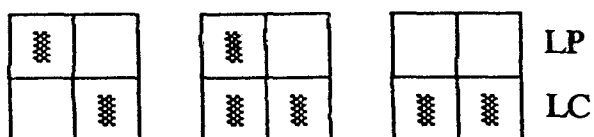
P2 : disparition de la ligne courante LC d'un segment apparu sur la ligne précédente LP.



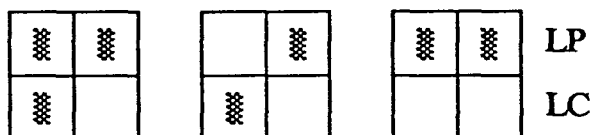
P3 : fin de séparation ou de réunion, entrée dans une forme ou poursuite de connexité 2D.



**P4** : apparition d'un début d'une réunion et poursuite de la connexité sur la ligne courante LC. La réunion est confirmée au passage à la place P3 ou de la place P5.



**P5** : apparition d'un début de séparation et poursuite de la connexité sur la ligne précédente LP. La séparation est confirmée au passage à la place P3 ou de la place P4.



## IV. CHAINAGE DYNAMIQUE INTERLIGNE.

### IV.1. Règle de chaînage dynamique.

Le chaînage dynamique est basé sur une règle importante et simple:

A tout nouveau segment identifié comme étant un "début" est assignée une nouvelle forme. A tout segment identifié soit à une "fin", à une "séparation", à une "réunion" ou à une "épaisseur" est assignée la même forme que celle assignée au segment qui lui est connexe sur la ligne précédente LP.

Pour appliquer cette règle, il faut connaître à tout moment ce qui s'est passé au niveau de la ligne précédente LP. Pour ce faire, nous construisons deux tableaux. L'un, noté  $T_{LP}$ , indique les formes repérées par des numéros auxquelles sont associés les segments de la ligne précédente LP dans l'ordre de leur apparition en abscisse croissante. L'autre, noté  $T_{LC}$ , contient les mêmes informations pour la ligne courante LC. On appelle rang R d'un segment, son numéro d'ordre d'apparition le long de la ligne considérée. Notons qu'une forme F peut contribuer à plusieurs segments de rangs différents sur la même ligne et également sur 2 lignes consécutives ( cf figure 33 ).

### IV.2. Apparition d'une forme.

Ainsi, lorsqu'un "début" est identifié, nous sommes en présence d'un segment ou d'un point qui n'existait pas à la ligne précédente. Nous lui associons une nouvelle forme, à laquelle on affecte le numéro d'ordre d'apparition de ce nouveau début par rapport aux débuts déjà identifiés au cours de la scrutation ligne par ligne. Ce numéro de forme est stocké dans le tableau  $T_{LC}$  au rang R correspondant à son ordre d'apparition sur la ligne courante. ( cf figure 34 )

Au cours de l'exploration des lignes successives, on teste la connexité des segments de la ligne courante LC par rapport à ceux de la ligne précédente LP, qui sont tous associés à des formes déjà numérotées. S'il y a connexité, le même numéro de forme est associé aux segments connexes de



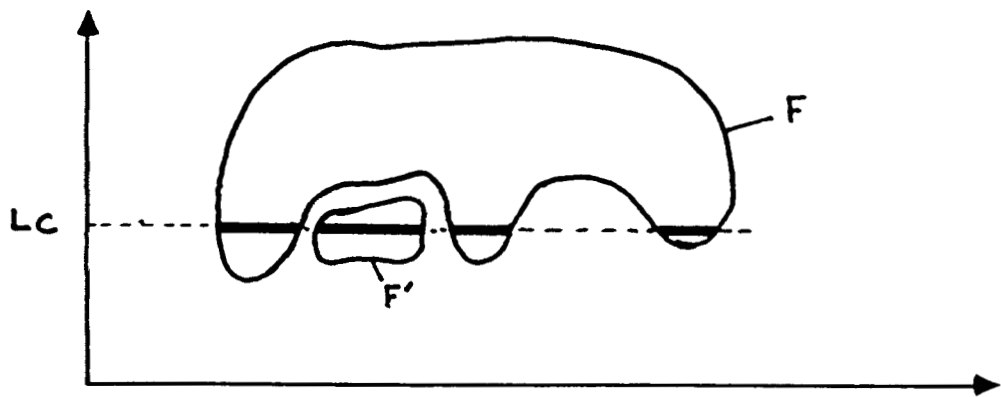


figure 33 : ligne précédente LP et ligne courante LC.

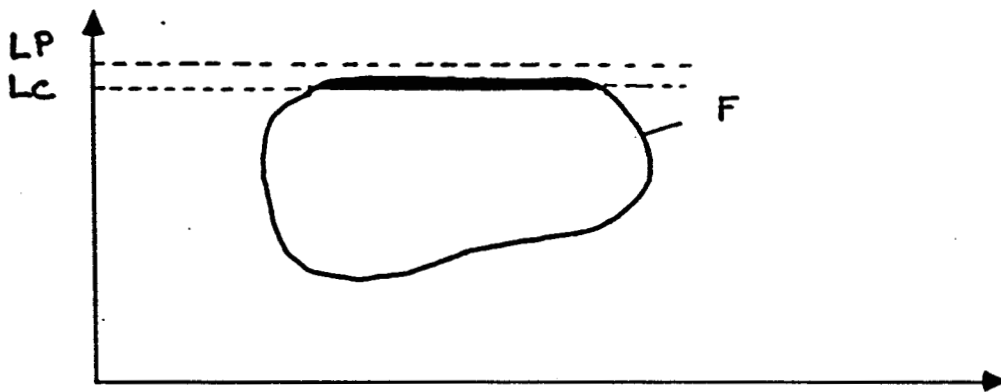


figure 34 : un exemple de début.

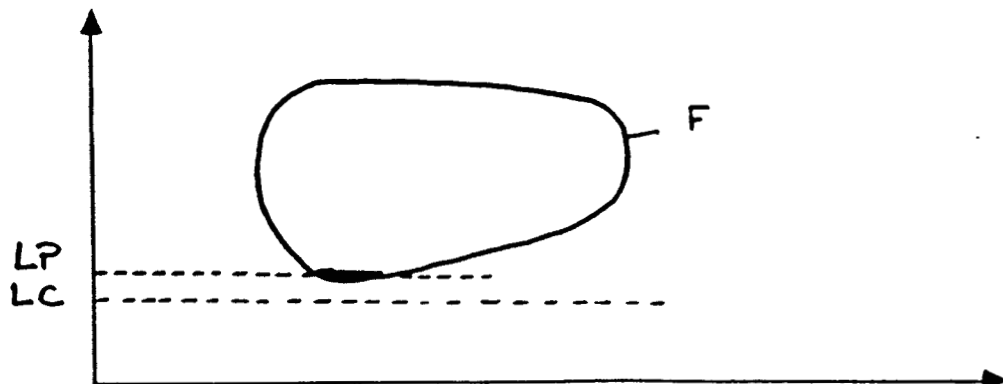


figure 35 : un exemple de fin.

la ligne précédente LP et de la ligne courante LC. Ce phénomène correspond à la configuration de "continuité de contour" ou "épaisseur".

#### IV.3. Apparition d'une fin.

Lorsqu'une "fin" est identifiée, c'est à dire lorsqu'un segment ou un point de la ligne précédente n'existe plus dans la ligne courante, le numéro de forme correspondant au rang du segment dans le tableau  $T_{LP}$  disparu entre les deux lignes consécutives est supprimé. Cette place libre dans le tableau  $T_{LC}$  contiendra le numéro de forme associé au prochain segment rencontré sur la ligne courante LC. ( cf figure 35 )

#### IV.4. Séparation et réunion.

Lorsqu'une "séparation" est identifiée, c'est à dire lorsqu'un segment ou un point de la ligne précédente LP se divise en deux segments ou deux points à la ligne courante LC, le numéro de forme correspondant au rang du segment considéré sur la ligne précédente occupe deux rangs dans le tableau  $T_{LC}$ . ( cf figure 36 )

Lorsqu'une "réunion" est identifiée, c'est à dire lorsque deux segments ou deux points de la ligne précédente se réunissent à la ligne courante, nous devons distinguer deux cas :

- les deux segments ou deux points ont le même numéro de forme.
- les deux segments ou deux points ont des numéros de forme différents.

Dans le cas où les segments ou les points proviennent de la même forme, c'est à dire lorsque le même numéro de forme leur est associé, ce numéro est simplement enregistré dans le tableau  $T_{LC}$  au rang où le segment "réunion" a été identifié. Dans le cas où les numéros de forme sont différents, nous devons définir le numéro de forme associé au segment ou au point qui réalise la réunion. Nous avons choisi de prendre le plus petit des deux numéros comme numéro de forme associé au segment "réunion". Ce numéro de forme prend place, dans le tableau  $T_{LC}$ , au rang correspondant à

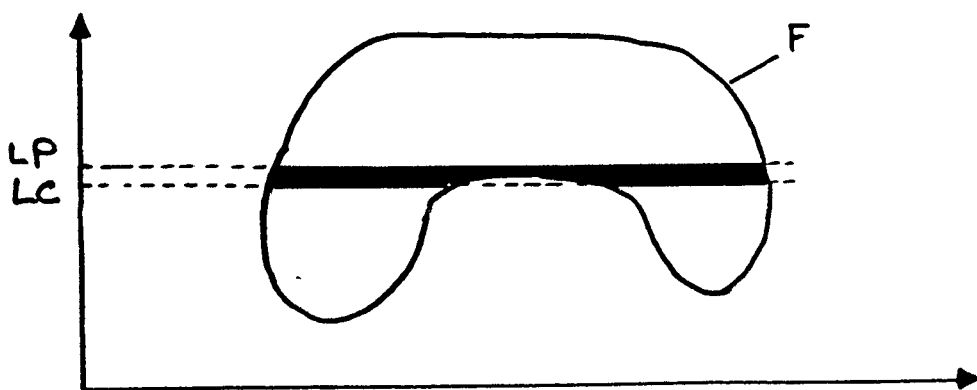


figure 36 : un exemple de séparation.

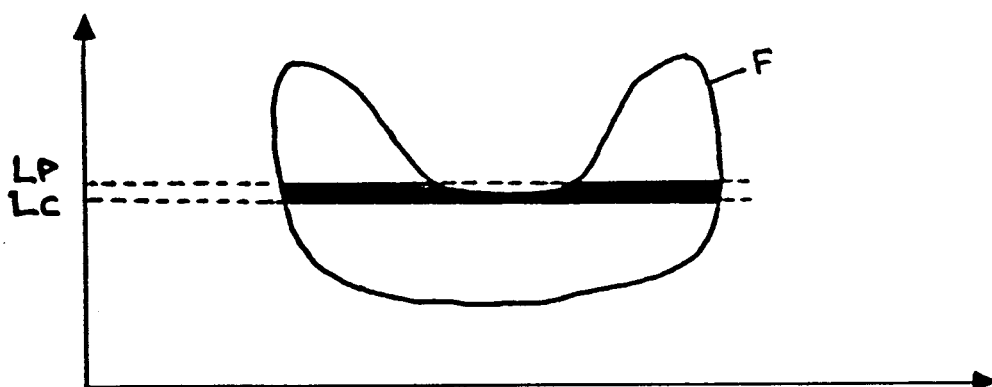


figure 37 : un exemple de réunion.

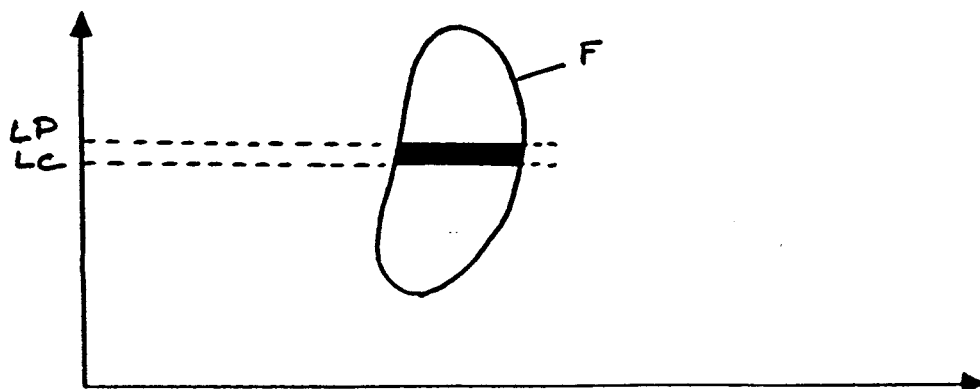


figure 38 : un exemple d'épaisseur.

son ordre d'apparition sur la ligne courante. Mais nous devons également changer le numéro de forme non retenu par le numéro de forme retenu dans tout le tableau  $T_{LP}$  ainsi que dans le tableau  $T_{LC}$  déjà construit. Cette procédure permet d'éviter d'avoir à réunir des segments ou points ayant des numéros de forme différents alors qu'on sait déjà qu'ils appartiennent à la forme. L'opération se réduit à une simple réunion de segments ou de points ayant le même numéro de forme. ( cf figure 37 )

## V. CONCLUSION.

La méthode de chaînage proposée présente l'avantage de reconstituer parfaitement, ligne après ligne, les contours des formes défilant sous la caméra d'un système de vision linéaire à partir des lignes de points de contour. Par une analyse de connexité dynamique, elle permet d'extraire d'une scène en défilement continu sous une caméra les contours qui délimitent chaque forme ou chaque objet observé.

Nous allons maintenant exploiter cette procédure séquentielle d'extraction des contours des objets pour les caractériser par des attributs métriques. Ces derniers doivent donc être calculés de manière dynamique, en respectant le rythme d'acquisition des lignes successives de l'image.

De plus, à ce changement de numéros de forme dans le tableau  $T_{LP}$  et le tableau  $T_{LC}$  s'ajoutent des changements au niveau du tableau où sont stockées et calculées toutes les caractéristiques des formes. En effet, le numéro de forme nous sert de pointeur dans un tableau. Dans ce tableau, nous stockons et calculons les attributs métriques des formes au fur et à mesure du chaînage dynamique. Cette procédure est abordée au chapitre suivant. Nous devons donc, dans le cas de réunion de segments ou de points ayant des numéros de forme différents, chaîner les attributs métriques correspondants à des formes à priori distinctes qui ont été finalement réunies au cours du chaînage. Ce chaînage au niveau de ce tableau se traduit par des transferts de données et des calculs souvent importants qui doivent perturber le moins possible le chaînage complet des lignes consécutives.

## **CHAPITRE V**

### **SYSTEME DE VISION LINEAIRE.**

Le chaînage dynamique des segments et points de contour présenté précédemment permet de déterminer le contour de chaque forme défilant sous la caméra au fur et à mesure de la détection des points de contour dans les image-lignes successives.

On pourrait envisager une mémorisation de ces contours sous la forme d'une description exhaustive indiquant les positions spatiales de chacun de leurs points élémentaires. Une telle démarche irait toutefois à l'encontre de l'idée générale de l'approche développée dans ce travail.

En effet, jusqu'à présent, nous avons surtout tenté de nous démarquer des procédures classiques de traitement des images matricielles en évitant de devoir mémoriser plus d'informations que celle contenues dans la ligne courante et une ou deux lignes consécutives.

C'est dans ce même esprit que nous abordons maintenant le calcul d'attributs métriques qui permettent de caractériser les formes extraites de la scène en défilement continu sous la caméra.

Cependant, l'implantation de ces procédures de caractérisation des formes est étroitement liée à l'architecture informatique du système de vision linéaire utilisé. Ce dernier est décrit au paragraphe suivant.

## I. SUPPORT MATERIEL.

Dans le cadre du développement de l'atelier flexible de confection à l'Université des Sciences et Techniques de Lille Flandres Artois, un tel système a été installé au dessus d'un convoyeur conduisant en continu le tissu d'un carrousel d'alimentation vers une table de découpe au laser. Ce système de vision est destiné à inspecter le tissu avant sa découpe. Cette inspection permettra de repérer les défauts dans le cas de tissus unis ou les motifs dans le cas de tissus imprimés. Les informations recueillies par le système de vision sont destinées à être transférées vers un système de placement dynamique. Il s'agira alors d'optimiser la découpe en tenant compte soit des défauts, soit de la position des motifs imprimés.

Pour cette application, le système Visionix est équipé d'une caméra Cyclope dotée d'une barrette de 2048 diodes. Les niveaux de gris sont codés sur 8 bits, de 0 pour le noir à 255 pour le blanc.

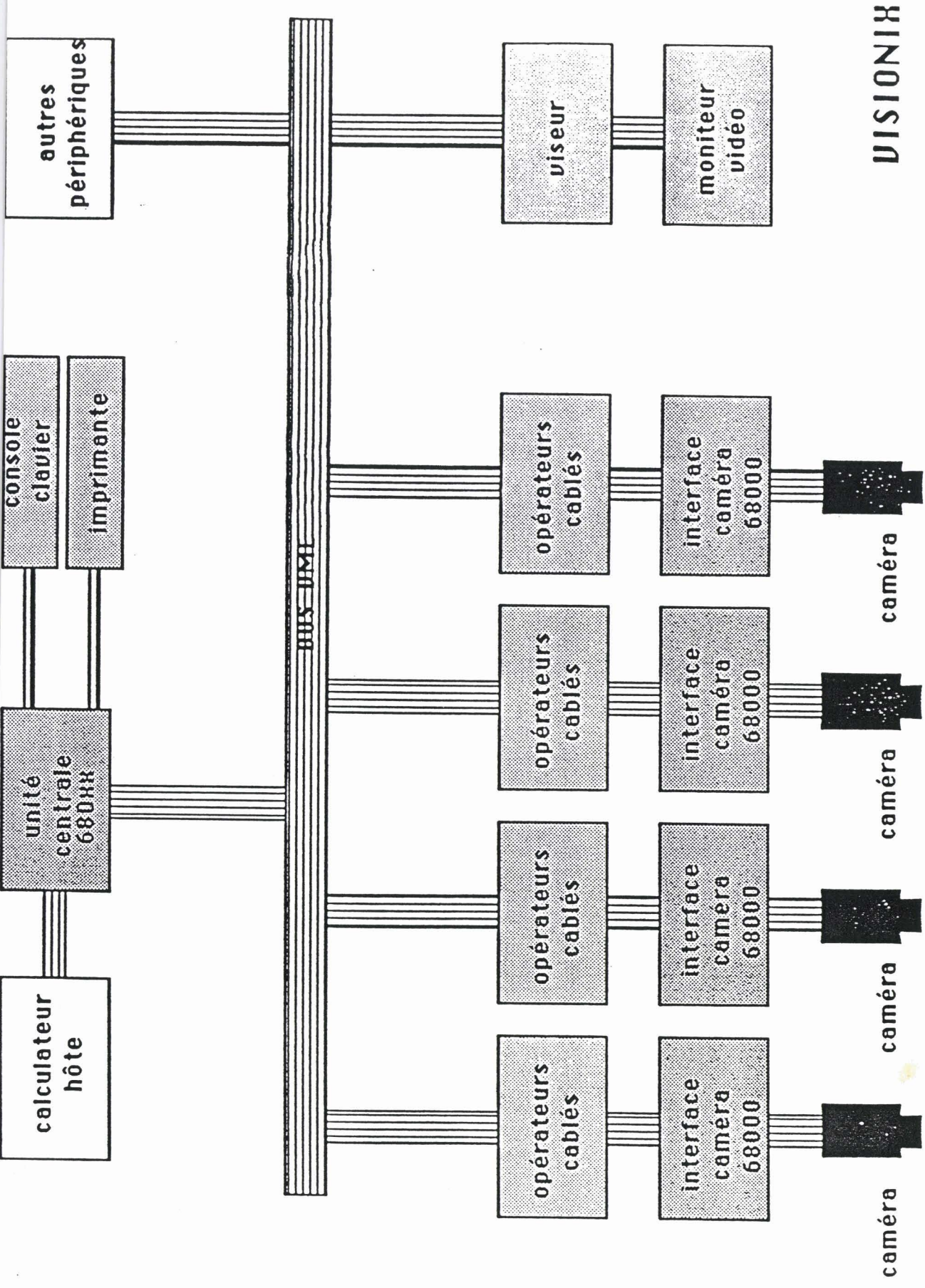
Le système comprend différents blocs fonctionnels:

a) Une unité centrale à base de processeurs de la famille 68000 à laquelle peuvent être adjointes des cartes d'entrées/sorties spécifiques et des extensions mémoires. Notre configuration est pourvue d'un microprocesseur 68000 ( 8 MHz ) et est dotée d'une extension mémoire de 1 méga-octets.

b) la caméra est reliée au système par le biais d'une unité de traitement spécifique. Cette unité d'interface, également équipée d'un processeur 68000, a trois grandes fonctions:

- le contrôle de la caméra par des registres de commande ou d'état,

- le stockage de la mesure de l'image-ligne, provenant de la caméra, dans 2 zones mémoire appelées la mémoire ligne A et la mémoire ligne B. Celles-ci sont en exclusion mutuelle, c'est à dire que si l'une est en mode lecture, l'autre est en mode écriture. De plus, l'unité dispose d'une mémoire de fond qui permet de stocker une image-ligne de référence.



**VISIONIX**

architecture du Visionix de Digital Design.

- les traitements de ces mesures, tels que la soustraction de la mémoire de fond en cours d'acquisition, ou le seuillage pour lequel les abscisses des points de franchissement du seuil sont stockées dans une mémoire F.I.F.O.

c) Une unité de visualisation soit des images provenant de la caméra, soit des images traitées. Cette unité permet de reconstituer l'image bidimensionnelle complète à partir des image-lignes successives.

La communication entre les différentes cartes se fait via le Bus VME, à l'initiative de l'unité centrale. Celle-ci est donc la carte maître du Bus VME. Les autres cartes, esclaves de l'unité centrale, ne communiquent avec la carte maître que par interruptions ou par stockage de données résultantes de traitements dans la zone mémoire commune. Un contrôleur d'interruptions gère les 8 niveaux d'interruptions hiérarchisées, masquables et vectorisées. Ces 8 niveaux d'interruptions sont, par ordre de priorité décroissante :

- niveau 0 : timer.
- niveau 1 : interruption provenant du maître du système.
- niveau 2 : interruption provenant du maître du système.
- niveau 3 : interruption provenant d'un module externe.
- niveau 4 : débordement mémoire F.I.F.O .
- niveau 5 : fin de cycle.
- niveau 6 : caméra prête.
- niveau 7 : interruption provenant d'un module externe.

On assigne le niveau 3 d'interruption sur le bus VME aux cartes d'interface caméra.



Pour le développement des logiciels d'application, nous disposons d'un système informatique sous système d'exploitation UNIX SYSTEM V. Il comporte des compilateurs et les bibliothèques C et Fortran, et un cross-assembleur 68000. Après développement, les logiciels sont téléchargés sur les différentes cartes cibles du Visionix.

Les logiciels sont développés en langages C et assembleur 68000. Compte tenu des impératifs temps réel, la majorité des programmes a été développée en langage assembleur 68000. L'architecture du système de vision étant spécialement conçue pour les traitements d'images ligne par ligne, il est vite apparu que la programmation en langage assembleur est parfaitement adaptée à l'exploitation optimale de la machine. Seuls les programmes gérant les relations Homme-Machine ont été développés en langage C.

## II. IMPLANTATION DES LOGICIELS.

### II.1. Détection des contours.

Le logiciel de détection des contours est implanté sur l'unité d'interface caméra. Cette tâche est activée par le programme principal implanté sur l'unité centrale par l'envoi d'une interruption sur le Bus VME.

Cette tâche est composée de deux parties :

. la première partie consiste à initialiser les registres de contrôle de la caméra (temps de pose, niveau du seuil s'il existe, état des F.I.F.O. et des registres), et à mettre la caméra en mode d'acquisition.

. pendant le temps de pose, l'image-ligne obtenue à l'acquisition précédente est stockée dans un tableau de dimensions 3 x 2048 bytes.

Dans ce tableau, sont ainsi stockées les image-lignes acquises aux instants  $T_{-3}$ ,  $T_{-2}$  et  $T_{-1}$  sachant que nous sommes en cours d'acquisition de l'image-ligne correspondant à l'instant  $T_0$ .

C'est sur ce tableau que nous appliquons l'opérateur de détection de contour. Nous avons introduit, dans le chapitre III, un opérateur morphologique sur une fenêtre 3 x 3 qui donne de meilleurs résultats que les opérateurs différentiels classiques du type Sobel et Prewitt. Cependant l'impératif temps réel ne nous permet pas d'utiliser ce type de fenêtre. En effet, son utilisation nécessiterait de nombreuses manipulations de données car il nous faudrait accéder au point considéré et à ses 8 voisins. C'est la raison pour laquelle nous avons opté pour la fenêtre cruciforme présentée au chapitre III qui permet de réduire le nombre de manipulations de données, tout en gardant une bonne qualité de détection. D'un point de vue temps de calcul, cette stratégie ne nécessite que l'accès au point considéré et à ses 4 plus proches voisins au lieu des 8 plus proches voisins dans le cas de la fenêtre 3 x 3.

Dans la fenêtre ainsi définie, que nous déplaçons le long du tableau, nous appliquons l'opérateur DOE. La détection mise en oeuvre pendant le temps de pose à l'instant  $T_0$ , s'applique donc aux image-lignes acquises aux instants  $T_{-3}$ ,  $T_{-2}$  et  $T_{-1}$  et le résultat du traitement est assigné à l'image-ligne de l'instant  $T_{-2}$ . Au fur et à mesure du déplacement de la fenêtre, les résultats sont stockés dans un tableau de dimensions 2 x 2046 en mémoire RAM. Les dimensions de ce tableau sont imposées par la nécessité de disposer 2 image-lignes binaires consécutives pour le chaînage dynamique. Ces calculs doivent donc être impérativement terminés en fin d'acquisition de l'image-ligne courante.

Dans notre application, cette tâche d'acquisition et de détection est prioritaire. En effet, le défilement continu des produits plats sous la caméra nous impose une acquisition ligne par ligne continue afin de ne perdre aucune information. La détection des contours, ainsi que les autres tâches implantées sur l'unité centrale s'effectuent donc pendant le temps de pose.

## II.2. Chaînage dynamique.

Grâce à la structure multiprocesseurs, pendant que la tâche d'acquisition et de détection s'exécute sur la carte d'interface caméra, le chaînage dynamique peut s'effectuer sur l'unité centrale. Pour cette tâche,

nous disposons donc d'un temps équivalent au temps de pose de la caméra puisque l'acquisition des image-lignes est prioritaire sur toutes les autres opérations, comme nous vous l'avons indiqué au paragraphe précédent.

Le chaînage dynamique utilise le tableau de dimensions  $2 \times 2046$  où sont stockées les lignes de points de contour obtenues aux instants  $T_{-3}$  et  $T_{-2}$ , sachant que nous sommes en cours d'acquisition de l'image-ligne correspondant à l'instant  $T_0$ .

La tâche du chaînage dynamique comprend en fait le chaînage des segments ou des points de contour proprement dit ainsi que la détermination des attributs métriques des formes associées à ces contours chaînés. Ce dernier point, étroitement associé et imbriqué au chaînage des contours est présenté en détail au paragraphe suivant.

### III. DETERMINATION DYNAMIQUE DES ATTRIBUTS METRIQUES.

Comme lors de la reconstruction des contours, ces attributs sont calculés de manière séquentielle, grâce à des procédures itératives qui permettent de les actualiser dès que les nouveaux éléments d'une ligne de points de contour sont assignés aux formes en cours de reconstitution. Ainsi, dès qu'un contour est complètement reconstitué, on disposera simultanément d'un certain nombre de paramètres caractérisant la forme correspondante. Ceux-ci peuvent être transmis à un système de reconnaissance de formes sur lequel seront implantés les procédures de décision en fonction de l'exploitation prévue de ces informations visuelles.

#### III.1. Notations et définitions.

Pour faciliter la présentation du calcul itératif des attributs métriques, on définit un segment de contour  $S_i$  par la ligne  $L_i$  à laquelle il appartient et par ses extrémités  $A_i$  et  $B_i$  ( cf figure 39 ).

Cette nouvelle notation se déduit aisément des notations introduites au chapitre précédent.

$A_i$  est en fait l'abscisse de la fenêtre d'observation  $W(X_0)$ ,  $B_i$  l'abscisse de la fenêtre d'observation  $W(X_{f+1})$  et  $L_i$  le numéro de la ligne à laquelle appartient le segment  $S_i$ .

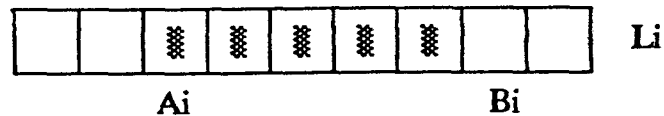


figure 39

Pour caractériser les formes défilant sous la caméra par des attributs métriques, nous utilisons les moments des contours extraits en temps réel. Ceux-ci sont calculés à partir de la définition générale du moment d'ordre  $p+q$  :

$$m_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x,y) x^p y^q dx dy$$

où  $f(x,y)$  est la densité au point de coordonnées  $(x, y)$ . Cette densité prend la valeur unité si le point de coordonnées  $(x, y)$  appartient à un contour et la valeur nulle autrement.

## III.2. Masse.

### III.2.1. Cas continu.

Soit un segment  $[l_0, l_f]$ , de longueur  $l$  et supposé sans épaisseur.  
( cf figure 40 )

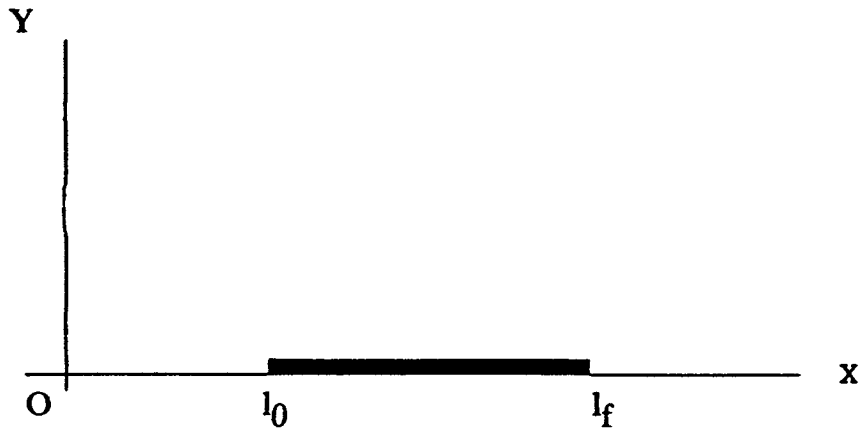


figure 40

La densité  $f(x,y)$  est définie par :

$$f(x,y) = 1, \text{ si } (x,y) \text{ appartient à } [l_0, l_f]$$

et

$$f(x,y) = 0, \text{ autrement.}$$

Dans ces conditions, la masse du segment est équivalente au moment d'ordre 0.

$$m_{00} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x,y) x^0 y^0 dx dy$$

$$m_{00} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x,y) dx dy$$

$$m_{00} = \int_{l_0}^{l_f} dx$$

$$m_{00} = l_f - l_0 = l$$

On voit très facilement que la longueur  $l$  est égale à la masse  $m$  du segment.

### III.2.2. Cas discret.

A partir de la remarque précédente, un segment  $S_i$  défini par  $(A_i, B_i, L_i)$  a pour masse  $m_i$ :

$$m_i = B_i - A_i$$

Donc, la masse totale  $M_{(N)}$  d'un contour constitué de  $N$  segments est égale à la somme des masses des segments le constituant :

$$M_{(N)} = \sum_{i=1}^N (B_i - A_i)$$

### III.2.3. Implantation.

La programmation de la fenêtre d'observation permet un accès direct aux valeurs  $A_i$  et  $B_i$  qui définissent chaque segment. Le calcul de la masse d'un segment est donc une distance et l'ajout de celle-ci à la masse du contour en cours de chaînage ne pose aucun problème.

## III.3. Centre de gravité.

### III.3.1. Cas continu.

Les coordonnées  $x_g$  et  $y_g$  du centre de gravité d'un segment, dans le repère  $(0, X, Y)$ , sont calculées à partir des moments d'ordre 1 et d'ordre 0:

$$x_g = \frac{m_{10}}{m_{00}} = \frac{\int_{l_0}^{l_f} x \, dx}{\int_{l_0}^{l_f} dx} = \frac{m}{2}$$

et

$$y_g = \frac{m_{01}}{m_{00}}$$

puisque le segment n'ayant pas d'épaisseur et étant situé sur l'axe OX, y est toujours nul.

### III.3.2. Cas discret.

Pour un segment  $(A_i, B_i, L_i)$  et de masse  $(B_i - A_i)$ , il est facile de calculer les coordonnées  $x_{G_i}$  et  $y_{G_i}$  de son centre de gravité, noté  $G_i$  :

$$x_{G_i} = \frac{(B_i + A_i - 1)}{2}$$

et

$$y_{G_i} = L_i$$

Ainsi les coordonnées  $X_G$  et  $Y_G$  du centre de gravité  $G$  d'un contour constitué de  $N$  segments se calculent sous la forme :

$$X_G = \frac{\sum_{i=1}^N \left[ (B_i - A_i) \cdot \frac{(B_i + A_i - 1)}{2} \right]}{\sum_{i=1}^N (B_i - A_i)}$$

et

$$Y_G = \frac{\sum_{i=1}^N \left[ (B_i - A_i) \cdot L_i \right]}{\sum_{i=1}^N (B_i - A_i)}$$

A partir de ces deux expressions des coordonnées du centre de gravité, nous déduisons les formules de récurrence nécessaires au calcul du centre de gravité au fur et à mesure que les segments constituant le contour apparaissent, ligne après ligne.

Pour l'abscisse du centre de gravité, on peut écrire :

$$X_G \sum_{i=1}^N (B_i - A_i) = \sum_{i=1}^{N-1} \left[ (B_i - A_i) \cdot \frac{(B_i + A_i - 1)}{2} \right] + (B_N + A_N) \cdot \frac{(B_N + A_N - 1)}{2}$$

Cette expression s'écrit sous la forme récurrente :

$$X_{G(N)} = X_{G(N-1)} + (B_N - A_N) \cdot \frac{(B_N + A_N - 1)}{2}$$

avec :

$$X_{G(N)} = X_G \cdot \sum_{i=1}^N (B_i - A_i)$$

et

$$X_{G(N-1)} = \sum_{i=1}^{N-1} \left[ (B_i - A_i) \cdot \frac{(B_i + A_i - 1)}{2} \right]$$

Pour l'ordonnée du centre de gravité, on a de la même manière :

$$Y_G \sum_{i=1}^N (B_i - A_i) = \sum_{i=1}^{N-1} \left[ (B_i - A_i) \cdot L_i \right] + (B_N - A_N) \cdot L_N$$



d'où :

$$Y_{G(N)} = Y_{G(N-1)} + (B_N - A_N) \cdot L_N$$

en posant :

$$Y_{G(N)} = Y_G \cdot \sum_{i=1}^N (B_i - A_i)$$

et

$$Y_{G(N-1)} = \sum_{i=1}^{N-1} [ (B_i - A_i) \cdot L_i ]$$

### III.3.3. Implantation.

Lorsqu'on chaîne un Nième segment à un contour, nous connaissons  $A_N$ ,  $B_N$  et  $L_N$ . Il suffit donc de calculer  $(B_N - A_N)(B_N + A_N - 1)/2$  et  $(B_N - A_N) L_N$  et d'ajouter ces valeurs aux coordonnées  $X_{G(N-1)}$  et  $Y_{G(N-1)}$  calculées à partir des N-1 segments précédemment chaînés.

Pour le cas particulier de l'identification d'une réunion de deux contours distincts P et Q, respectivement de masses  $M_P$  et  $M_Q$  et de centres de gravité de coordonnées  $(X_{GP}, Y_{GP})$  et  $(X_{GQ}, Y_{GQ})$ , on calcule le centre de gravité global obtenu par fusion des contours P et Q sous la forme :

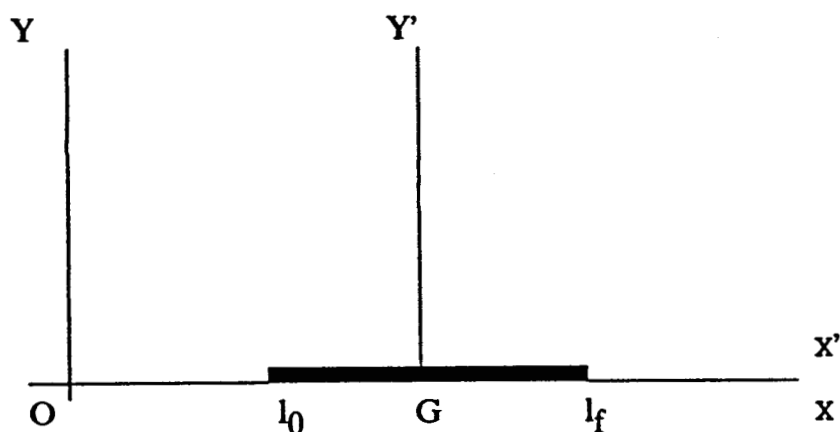
$$X_G = \frac{(M_P X_{GP} + M_Q X_{GQ})}{(M_P + M_Q)}$$

$$Y_G = \frac{(M_P Y_{GP} + M_Q Y_{GQ})}{(M_P + M_Q)}$$

### III.4. INERTIES

#### III.4.1. Cas continu.

Dans le repère  $(O, X, Y)$ , on considère le segment  $[l_0, l_f]$  supposé sans épaisseur et de masse  $m$ . Nous allons calculer :



- le moment central d'inertie  $I_{gy}$  par rapport à l'axe  $GY'$ ,
- le moment central d'inertie  $I_{gx}$  par rapport à l'axe  $GX'$ ,
- le produit central d'inertie  $I_{gxy}$ .

Les moments et produits d'inertie  $I_{gy}$ ,  $I_{gx}$  et  $I_{gxy}$  se calculent respectivement à partir des moments  $m_{10}$ ,  $m_{01}$  et  $m_{11}$ . Nous obtenons :

$$I_{gy} = m_{10} = \int_{-1/2}^{+1/2} f(x, y) x^2 dx$$

$$I_{gy} = \int_{-1/2}^{+1/2} x^2 dx = \left[ \frac{x^3}{3} \right]_{-1/2}^{+1/2}$$

$$I_{gy} = \frac{l^3}{12}$$

$$I_{gx} = m_{01} = 0 \text{ et}$$

$$I_{gxy} = m_{11} = 0 \text{ puisque } y=0.$$

Ces moments et produits d'inertie peuvent être recalculés par rapport au centre d'inertie  $G_{(N)}$  de l'ensemble des  $N$  segments chaînés du contour, en appliquant le théorème de Huyghens [POL-74]. Nous obtenons donc :

$$- I_{Gy} = I_{gy} + m \cdot X_{Gg}^2$$

$$- I_{Gx} = I_{gx} + m \cdot Y_{Gg}^2$$

$$- I_{Gxy} = I_{gxy} + m \cdot X_{Gg} \cdot Y_{Gg}$$

où ;

$X_{Gg}$  est la projection sur l'axe  $OX$  du segment joignant le centre de gravité  $G_{(N)}$  et le centre de gravité  $g_i$  du segment  $S_i$ ,

et  $Y_{Gg}$  est la projection sur l'axe  $OY$  du segment joignant le centre de gravité  $G_{(N)}$  et le centre de gravité  $g_i$  du segment  $S_i$ .

### III.4.2. Cas discret.

Nous pouvons appliquer les résultats obtenus dans le cas continu au cas discret. Soit un contour constitué de  $N$  segments  $S_i$ , définis par  $(A_i, B_i, L_i)$ , de masses  $m_i = (B_i - A_i)$  et de centres de gravité  $g_i$  de coordonnées  $(x_{gi}, y_{gi})$ . Ses moments et produits d'inertie centrés par rapport au centre de gravité  $G_{(N)}$  de coordonnées  $(X_{G(N)}, Y_{G(N)})$  sont :

$$I_{GY(N)} = \frac{\sum_{i=1}^N (B_i - A_i)^3}{12} + \sum_{i=1}^N \left[ (B_i - A_i) (X_{g_i G(N-1)})^2 \right]$$

$$I_{GX(N)} = \sum_{i=1}^N \left[ (B_i - A_i) (Y_{g_i G(N-1)})^2 \right]$$

$$I_{GXY(N)} = \sum_{i=1}^N \left[ (B_i - A_i) \cdot X_{g_i G(N-1)} \cdot Y_{g_i G(N-1)} \right]$$

Comme dans le cas du calcul du centre de gravité, nous pouvons obtenir des formules de récurrence.

En posant,  $X_{g(N)G(N-1)}$  et  $Y_{g(N)G(N-1)}$  les projections de la distance entre le centre de gravité  $G_{(N-1)}$  du contour constitué de  $N-1$  premiers segments chaînés et le centre de gravité  $g_{(N)}$  du Nième segment chaîné, on obtient les moments et produits centraux d'inertie par rapport aux axes  $G_{(N)} X'$  et  $G_{(N)} Y'$  :

$$I_{GY(N)} = I_{GY(N-1)} + \frac{(B_N - A_N)^3}{12} + (B_N - A_N) (X_{g_N G(N-1)})^2$$

avec :

$$I_{GY(N-1)} = \frac{\sum_{i=1}^{N-1} (B_i - A_i)^3}{12} + \sum_{i=1}^{N-1} \left[ (B_i - A_i) (X_{g_i G(N-2)})^2 \right]$$

Le moment central d'inertie par rapport à l'axe  $G_{(N)} X'$  prend la forme :

$$I_{GX(N)} = I_{GX(N-1)} + (B_N - A_N) (Y_{g_N G_{(N-1)}})^2$$

avec :

$$I_{GX(N-1)} = \sum_{i=1}^{N-1} \left[ (B_i - A_i) (Y_{g_i G_{(N-1)}})^2 \right]$$

Pour le produit d'inertie, nous avons :

$$I_{GXY(N)} = I_{GXY(N-1)} + (B_N - A_N) X_{g_N G_{(N-1)}} Y_{g_N G_{(N-1)}}$$

avec :

$$I_{GXY(N-1)} = \sum_{i=1}^{N-1} \left[ (B_i - A_i) X_{g_i G_{(N-2)}} Y_{g_i G_{(N-2)}} \right]$$

#### II.4.3. Implantation.

Connaissant les caractéristiques du segment  $S_N$  chaîné, il faut calculer :

$$\frac{(B_N - A_N)^3}{12} + (B_N - A_N) (X_{g_N G_{(N-1)}})^2$$

$$(B_N - A_N) (Y_{G_{N(N-1)}})^2$$

et

$$(B_N - A_N) (X_{G_{N(N-1)}}) \cdot (Y_{G_{N(N-1)}})$$

La remise à jour des moments et produits d'inertie se déroule de la même manière que pour les attributs précédents.

Pour le cas d'une réunion de deux contours P et Q distincts, il faut intégrer les inerties du contour P aux inerties du contour Q en appliquant le théorème de Huyghens.

Les résultats de cette réunion sont :

$$I_{GX} = I_{GXQ} + M_Q Y_{PQ}^2,$$

$$I_{GY} = I_{GYQ} + M_Q X_{PQ}^2$$

et

$$I_{GXY} = I_{GXYQ} + M_Q X_{PQ} Y_{PQ}$$

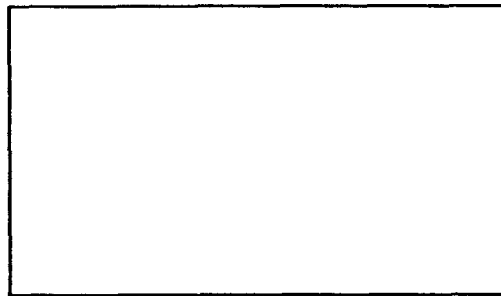
où  $X_{PQ}$  et  $Y_{PQ}$  sont respectivement les projections sur les axes OX et OY du segment joignant les centres de gravité  $G_P$  et  $G_Q$ .

### III.5. Rectangle exinscrit.

Une autre technique de caractérisation de forme, souvent utilisée et qui se détermine facilement, est le rectangle exinscrit dont les cotés sont parallèles et perpendiculaires aux axes de l'image. C'est le rectangle de surface minimum pouvant contenir complètement le contour considéré. Les coordonnées de ce rectangle se déduisent facilement à partir des définitions des segments constituant le contour chaîné.

Ce rectangle se construit à partir de deux points de coordonnées  $(XMIN, YMIN)$  et  $(XMAX, YMAX)$ .

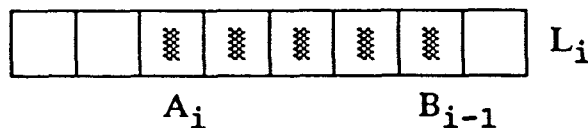
$(XMIN, YMIN)$



$(XMAX, YMAX)$

En reprenant la définition d'un segment  $S_i$  par les 3 paramètres :

$(A_i, B_i, L_i)$



Un contour constitué de  $N$  segments peut être caractérisé par :

-  $XMIN$ , la plus petite valeur que peut prendre  $A_i$  pour les  $N$  segments,

ou encore,

-  $XMIN = \min \{ A_i ; i \text{ appartenant à } [ 1, N ] \}$ .

-  $XMAX$ , la plus grande valeur que peut prendre  $B_{i-1}$  pour les  $N$  segments,

ou encore,

-  $XMAX = \max \{ B_{i-1} ; i \text{ appartenant à } [ 1, N ] \}$ .

-  $YMIN$ , le numéro de la ligne où est apparu le premier segment identifié comme le début du contour.

-  $YMAX$ , le numéro de la ligne où est apparu le dernier segment identifié comme étant la fin du contour.

Dans le cas où le contour est la réunion de deux contours  $P$  et  $Q$ ,  $YMIN$  du contour obtenu par fusion sera la valeur minimale du  $YMIN$  du contour  $P$  et du  $YMIN$  du contour  $Q$ .

Nous obtenons ainsi les deux points élémentaires définissant le rectangle exinscrit de ce contour.

Pour certains contours, ce paramètre peut être très significatif et intéressant. Il peut déboucher sur des calculs de taux d'occupation du rectangle exinscrit par le contour. Ce paramètre nécessite la détermination de la surface intérieure du contour.

### III.6. Aire.

Si l'on désire respecter la technique de calcul séquentiel des paramètres caractéristiques des formes définies par leur contour, on ne peut calculer la véritable surface intérieure au contour. En effet, dans le cas de formes complexes, nous avons montré, au chapitre IV, que le remplissage des formes n'était pas un problème trivial.



Toutefois, sous certaines hypothèses, telles que l'absence de formes concaves, un remplissage séquentiel peut être envisagé grâce à une procédure simple compatible avec les contraintes imposées.

La figure xxx indique les limites de la méthode de calcul de la surface. Notons cependant que, même dans les cas de formes non convexes, le paramètre ainsi déterminé peut être utilisé pour effectuer une discrimination entre différents type de formes cf figure .

Cette surface se détermine, par récurrence, par l'ajout des surfaces élémentaires  $SF_i$  calculées pour chaque ligne par :

$$SF_i = \max (B_{i-1}) - \min (A_i)$$

#### IV. TIMING. ASPECT TEMPS REEL.

Le temps de pose a été fixé à 100 millisecondes. Cette valeur correspond au temps de pose nécessaire à une bonne intégration des photosensibles, donc à une bonne dynamique du signal vidéo, compte tenu des conditions particulières de notre application où l'uniformité de l'éclairage n'est pas garantie sur la grande largeur contrôlée. Le convoyeur alimentant la table de découpe laser avançant à une vitesse d'environ 0,6 mètre par minute, cette période de 100 ms permet d'obtenir une résolution spatiale longitudinale de l'ordre du millimètre. Cette résolution est toute à fait cohérente avec la résolution transversale qui est également de l'ordre du millimètre avec la barrette de 2048 photodiodes de la caméra Cyclope qui couvre une scène de 2 mètres de largeur.

Ce temps de pose est un paramètre fondamental du fonctionnement de notre système car l'exécution de toutes les tâches de traitement jusqu'à l'extraction des caractéristiques des formes doivent toutes être exécutées durant ce temps de pose.

Les tâches implantées sur la carte interface caméra qui fixent les paramètres de contrôle de la caméra et qui détectent les contours sur 3 image-lignes consécutives et stockent les résultats dans un tableau,

nécessitent un temps de traitement de l'ordre de 95 millisecondes. Ce temps a été obtenu avec un processeur 68000 cadencé à 8 Méga Hertz. Nous pouvons donc spéculer sur un temps de traitement correspondant au tiers de celui actuellement obtenu si nous utilisons un processeur 68020. En effet, celui-ci à une fréquence de d'horloge de 16,67 Méga Hertz, le cycle bus du 68020 s'effectue en 180 nanosecondes tandis que celui du 68000 ou 68010 s'effectue en 240 nanosecondes mais surtout, et c'est l'explication d'un tel temps d'exécution, le 68020 est un véritable processeur 32 bits. Ce processeur, par son architecture, nous offre des possibilités de programmation plus étendues avec, entre autres, un registre de mode d'adressage plus important ( 18 types d'adressage distincts contre 14 pour le 68000 ).

Lors de la mise au point de la tâche implantée sur l'unité centrale, le temps d'exécution du chaînage dynamique proprement dit oscille entre 20 et 30 millisecondes, suivant le nombre de segments ou points de contour à chaîner. Pour atteindre ce temps de calcul, le chaînage a été développé en langage assembleur. La géométrie, les dimensions et le codage de la fenêtre d'observation définie au chapitre IV se prêtent relativement bien à cette programmation. Pour atteindre ces temps d'exécution très faibles, surtout au niveau des calculs des moments et produits d'inertie, il nous a fallu développer des macro-instructions en langage assembleur permettant des opérations de division 32 bits par 32 bits ou encore des opérations de multiplication 32 bits par 16 bits ou 32 bits par 32 bits, opérations absentes du jeu d'instructions du 68000.

Pour l'instruction MULU ( multiplication non signée 16 par 16 bits et résultat sur 32 bits ), le 68010 met 40 cycles horloge contre 70 pour le 68000. L'utilisation d'un processeur 68020 permettrait d'abaisser de façon spectaculaire le temps d'exécution de cette phase qui est surtout constituée de calculs arithmétiques. En effet, compte tenu de l'interface co-processeur dont est muni le 68020, l'association du 68020 avec un processeur de gestion mémoire comme le PMMU 68851 et un processeur de calcul arithmétique et scientifique comme le FPCP 68881 fait de celui-ci un "super" microprocesseur disposant des fonctionnalités des deux boîtiers comme s'il n'en faisait qu'un, à la différence des circuits périphériques classiques. Le jeu d'instructions et les registres internes de ce "super" microprocesseur est l'ensemble des jeux

d'instructions et des registres internes des circuits mis en commun. Le dialogue entre les boîtiers est complètement transparent au programmeur.

En conclusion, nous pouvons prévoir des temps de traitement nettement inférieurs à ceux présentés ici, en changeant de processeur, ce qui ne nécessiterait que quelques modifications mineures des logiciels.

## CONCLUSION GENERALE.

Nous vous avons présenté un système d'inspection automatique en temps réel par caméra linéaire de produits plats en défilement continu. Ce système, mettant en oeuvre des techniques basées sur la morphologie mathématique appliquée aux images en niveau de gris et un chaînage dynamique pour la reconstruction des contours détectés, n'intègre aucune connaissance à priori du produit à inspecter. Il permet d'extraire en temps réel des caractéristiques très utiles pour une reconnaissance de forme.

Le chapitre I rappelle l'importance croissante de l'inspection automatique des produits dans une chaîne de production. En effet, un contrôle visuel par un système de vision permet d'éliminer en cours de fabrication les produits défectueux et ceci, en garantissant des performances constantes du contrôle. Mais un tel système demande des soins particuliers qu'en au choix et à la qualité du système d'éclairage. Des investissements souvent lourds et des modifications matérielles importantes sur la chaîne de production ne pouvant être envisagés, nous devons faire appel à des techniques de traitement de l'image à la fois robustes par rapport aux variations de l'éclairage et rapides afin de respecter les exigences du temps réel.

Le chapitre II présente les techniques classiques de segmentation des images recueillies par un système de vision à caméra linéaire. Les techniques de seuillage, les techniques soustractives ou les techniques mettant en oeuvre des opérateurs différentiels, se révèlent insatisfaisantes compte tenu aux exigences de robustesse par rapport aux variations spatiales et temporelles de l'éclairage de l'absence complète de connaissances à priori de certaines caractéristiques des produits à inspecter et des contraintes du temps réel. Nous nous sommes donc orientés vers des techniques basées sur la Morphologie Mathématique.

Les opérations de base de la Morphologie Mathématique appliquée aux images binaires et aux images en niveaux de gris sont présentées au chapitre III. De l'étude ces opérations appliquées aux images en niveaux de gris nous dégagons une technique pour éliminer le bruit perturbant les image-lignes afin de détecter parfaitement les formes présentes dans la scène en mouvement sous la caméra, sans aucune connaissance à priori. Cette technique, robuste par rapport aux variations de l'éclairage, a été développée avec pour objectif permanent la minimisation du temps d'exécution afin de satisfaire aux exigences du temps réel.

Les contours détectés dans les image-lignes par cette technique sont chaînés par une méthode présentée au chapitre IV, basée sur un chaînage dynamique permettant de reconstituer les formes. Cette reconstruction s'effectue au fur et à mesure de la détection, contour après contour, ligne par ligne et toujours en temps réel. Les caractéristiques des formes rencontrées sont mises à jour au fur et à mesure du chaînage dynamique des contours détectés. Ces caractéristiques permettront de reconnaître les formes en différé.

Les calculs et les mises à jour en temps réel des caractéristiques des formes telles que la masse, le centre de gravité, les moments d'inertie sont présentés au chapitre V. L'implantation de l'ensemble de ces techniques sur un système de vision linéaire à architecture multi-processeurs intégré à l'atelier flexible de confection du Centre d'Automatique de Lille a permis d'évaluer les performances de ce système d'inspection automatique.

Les algorithmes ont été développés et implantés avec le souci constant de minimiser les temps de traitement à toutes les étapes tout en assurant une bonne détection des formes présentes. Dans l'état actuel du projet, le temps de traitement global est de 100 millisecondes par ligne. Ceci correspond en fait au temps de pose nécessaire à l'acquisition d'une image-ligne de la scène à inspecter qui, rappelons-le, est de grande largeur ( de l'ordre de 2 mètres ), en mouvement de l'ordre du mètre par minute et éclairée par la lumière de l'atelier.

Ce temps de traitement des image-lignes peut être réduit et ceci de deux manières. La première, la plus simple, consiste à doter le système de

vision, construit autour d'un bus VME, de cartes à microprocesseurs plus puissants comme le 68020 ou le 68030. La seconde consiste à utiliser des opérateurs cablés.

*Un tel système d'inspection par vision linéaire peut trouver sa place dans tout processus de contrôle visuel de produits plats en défilement. La robustesse des procédures permet de fixer les paramètres de réglage à des valeurs standard qui répondent à une grande variété de situations.*

L'intégration de ce système dans un processus de production est particulièrement aisée du fait qu'il nécessite que très peu d'installations matérielles et qu'il délivre en temps réel, sous forme standardisée, un grand nombre de paramètres classiques qui peuvent être repris par un système de reconnaissance de forme indépendant.

## CONCLUSION GENERALE.

Nous vous avons présenté un système d'inspection automatique en temps réel par caméra linéaire de produits plats en défilement continu. Ce système, mettant en oeuvre des techniques basées sur la morphologie mathématique appliquée aux images en niveau de gris et un chaînage dynamique pour la reconstruction des contours détectés, n'intègre aucune connaissance à priori du produit à inspecter. Il permet d'extraire en temps réel des caractéristiques très utiles pour une reconnaissance de forme.

Le chapitre I rappelle l'importance croissante de l'inspection automatique des produits dans une chaîne de production. En effet, un contrôle visuel par un système de vision permet d'éliminer en cours de fabrication les produits défectueux et ceci, en garantissant des performances constantes du contrôle. Mais un tel système demande des soins particuliers qu'en au choix et à la qualité du système d'éclairage. Des investissements souvent lourds et des modifications matérielles importantes sur la chaîne de production ne pouvant être envisagés, nous devons faire appel à des techniques de traitement de l'image à la fois robustes par rapport aux variations de l'éclairage et rapides afin de respecter les exigences du temps réel.

Le chapitre II présente les techniques classiques de segmentation des images recueillies par un système de vision à caméra linéaire. Les techniques de seuillage, les techniques soustractives ou les techniques mettant en oeuvre des opérateurs différentiels, se révèlent insatisfaisantes compte tenu aux exigences de robustesse par rapport aux variations spatiales et temporelles de l'éclairage de l'absence complète de connaissances à priori de certaines caractéristiques des produits à inspecter et des contraintes du temps réel. Nous nous sommes donc orientés vers des techniques basées sur la Morphologie Mathématique.

Les opérations de base de la Morphologie Mathématique appliquée aux images binaires et aux images en niveaux de gris sont présentées au chapitre III. De l'étude ces opérations appliquées aux images en niveaux de gris nous dégagons une technique pour éliminer le bruit perturbant les image-lignes afin de détecter parfaitement les formes présentes dans la scène en mouvement sous la caméra, sans aucune connaissance à priori. Cette technique, robuste par rapport aux variations de l'éclairage, a été développée avec pour objectif permanent la minimisation du temps d'exécution afin de satisfaire aux exigences du temps réel.

Les contours détectés dans les image-lignes par cette technique sont chaînés par une méthode présentée au chapitre IV, basée sur un chaînage dynamique permettant de reconstituer les formes. Cette reconstruction s'effectue au fur et à mesure de la détection, contour après contour, ligne par ligne et toujours en temps réel. Les caractéristiques des formes rencontrées sont mises à jour au fur et à mesure du chaînage dynamique des contours détectés. Ces caractéristiques permettront de reconnaître les formes en différé.

Les calculs et les mises à jour en temps réel des caractéristiques des formes telles que la masse, le centre de gravité, les moments d'inertie sont présentés au chapitre V. L'implantation de l'ensemble de ces techniques sur un système de vision linéaire à architecture multi-processeurs intégré à l'atelier flexible de confection du Centre d'Automatique de Lille a permis d'évaluer les performances de ce système d'inspection automatique.

Les algorithmes ont été développés et implantés avec le souci constant de minimiser les temps de traitement à toutes les étapes tout en assurant une bonne détection des formes présentes. Dans l'état actuel du projet, le temps de traitement global est de 100 millisecondes par ligne. Ceci correspond en fait au temps de pose nécessaire à l'acquisition d'une image-ligne de la scène à inspecter qui, rappelons-le, est de grande largeur ( de l'ordre de 2 mètres ), en mouvement de l'ordre du mètre par minute et éclairée par la lumière de l'atelier.

Ce temps de traitement des image-lignes peut être réduit et ceci de deux manières. La première, la plus simple, consiste à doter le système de



vision, construit autour d'un bus VME, de cartes à microprocesseurs plus puissants comme le 68020 ou le 68030. La seconde consiste à utiliser des opérateurs cablés.

Un tel système d'inspection par vision linéaire peut trouver sa place dans tout processus de contrôle visuel de produits plats en défilement. La robustesse des procédures permet de fixer les paramètres de réglage à des valeurs standard qui répondent à une grande variété de situations.

L'intégration de ce système dans un processus de production est particulièrement aisée du fait qu'il nécessite que très peu d'installations matérielles et qu'il délivre en temps réel, sous forme standardisée, un grand nombre de paramètres classiques qui peuvent être repris par un système de reconnaissance de forme indépendant.

## BIBLIOGRAPHIE.

- [ABD-79] : ABDOU I.E., PRATT W.K.  
 Quantitative design and evaluation of enhancement/thresholding edge detections  
 Proc.IEEE, vol 67, n°5, pp 753-763, may 1979.
- [ACK-81] : ACKLAND B.D. ET WESTE N.H.  
 The edge flag algorithm. A fill method of raster scan displays.  
 I.E.E.E. Trans. Comp. Vol C-30, pp 41-48, Jan 1981.
- [AME-87] : AMEZIANE M.  
 Segmentation des images numériques par seuillages multiples.  
 Application à la découpe automatique dans les ateliers flexibles.  
 Thèse de Doctorat, Lille, 1987.
- [AME-86] : AMEZIANE M, POSTAIRE J.G.  
 Etiquetage dynamique pour l'extraction de régions par suivi de contour.  
 Congrès AFCET/IASTED, Identification et Reconnaissance des formes, Toulouse, Tome 2, pp 801-818, 1986.
- [AME-85] : AMEZIANE M., POSTAIRE J.G., BONNET P.  
 Vision applied to a cutting process in textile industry.  
 7th International Conference on Automated Inspection and Product Control. Birmingham. 1985.
- [AME-85] : AMEZIANE M, POSTAIRE J.G., BONNET P.  
 La vision artificielle appliquée au contrôle de la qualité et à l'automatisation de la découpe dans l'industrie textile.  
 Confection 2000, pp 108-113, 1985.

[ARN-78] : ARNAUD J.M., BARBA D, DUCHENE G.

Rapport ANVAR n°4, 60 pages, 1978.

[ASH-78] : ASHKAN G.P., MODESTINO J.W.

The contour extraction problem with biomedical applications.

C.G.I.P. 7, pp 331-355, 1978.

[BAJ-84] : BAJON J., CATTOEN M., KIM S.D.

Système et méthode d'extraction de caractéristiques d'images binaires pour les applications en robotique.

4 ième congrés AFCET/INRIA. Reconnaissance des formes et Intelligence Artificielle. Paris, Tome 1, pp 367-382, Jan 1984.

[BAJ-79] : BAJON J., CATTOEN M., RAKHODAI I.

Extracteur rapide de caractéristiques d'images vidéo destiné à la robotique.

2 ième congrés AFCET/INRIA. Toulouse, 1979.

[BAR-84] : BARRETT W.A.

An iterative algorithm for mutiple threshold detection.

Division of computer research and technology national institutes of health Bethesda, Maryland, pp 273-278, 1984.

[BAS-79] : BASSEVILLE M.

Détection de contours : méthodes et études comparatives.

Annales Télécommunications, tome 34, 1979.

[BON-87] : BONNET P.

Cours de traitement d'images. D.E.A., Centre d'automatique de Lille, 1987.

[BOU-88] : BOUTRY F.

Analyse de scènes dynamiques par représentation espace-temps.

Application à la conduite automatique de véhicules guidés.

Thèse de l'Université, Lille, 1988.

[BOU-85] : BOUTRY F.

Analyse du mouvement par squelettisation.  
Rapport de D.E.A. 1985.

[CAT-85] : CATTOEN M.

Conception et réalisation d'une machine de vision achromique et  
trichromique.  
Thèse d'Etat. I.N.P Toulouse. 1985.

[CHI-74] : CHIEN Y.P, FU K.S.

A decision fct. method for boundary detection.  
C.G.I.P. 3, 2, pp 125-140, 1974.

[COS-85] : COSTER M., CHERMANT J.L.

Précis d'analyse d'images.  
Ed CNRS, 1985.

[CRI-85] : CRIMMINS T.R., BROWN W.M.

Image algebra and automatic shape recognition.  
IEEE Trans Aerosp. Electron. Syst., vol AES-21, pp 60-69, Jan 1985.

[CRO-77] : CROW F.C.

Shadow algorithm for computer graphics.  
ACM SGGGRAPH Computer Graphics, vol 2, n°2, pp 242-248, 1977.

[DAN-81] : DANIELSSON P.E., LEVIALDI S.

Computer architectures for pictorial information systems.  
IEEE Comput. Mag., pp 53-67, Nov 1981.

[DEL-87] : DELEPLANQUE H.

Segmentation des images numériques par seuillage adapté. Application  
à l'automatisation du repiquage en agriculture.  
Thèse de Doctorat, Lille, 1987.

[DES-80] : DESSIMOZ J.D.

Specialized edge-trackers for contour extraction and line thinning.  
Signal Processing, Vol 2, pp 71-73, 1980.

[DEU-78] : DEUTSCH E.S.,FRAM J.R.

A quantitative study of the orientation bias of some edge detector schemes.

IEEE, Trans. Comp., vol C-27, n°3, pp 205-213, march 1978.

[DIS-82] : DISTANTE A., VENEZAINI N.

A two-pass filling algorithm for raster graphics.

C.G.I.P. 20, pp 288-295, 1982.

[DOU-88] : DOUCHEZ M.C.

Extraction de la composante éclaircissement por la segmentation des images numériques.

DEA USTLFA. Juin 1988.

[DUD-73] : DUDA R.O., HART P.E.

Pattern Calssification and Scene Analyse.

Wiley, N.Y. p 271, 1973.

[DUD-76] : DUDANI S.A.

Region extraction using boundary following.

Pattern Recognition and Artificial Intelligence.

Academic Press, N.Y., pp 216-222, 1976.

[DUF-79] : DUFF M.

Parallel processors for digital image processing.

Advances in Digital Image Processing, P. Stucki, Ed. New York :

Plenum pp 265-279, 1979.

[DUP-89] : DUPLOUY S. POSTAIRE J.G.

Un système d'inspection automatique pour la Blanchisserie Industrielle.

Rapport de fin de contrat MRT. Mars 1989.

[FRA-75] : FRAM J.R., DEUTSCH E.S.

On the evaluation of edge detector schemes and their comparison with human performance.

IEEE, Trans. Comp. vol C-24, n°6, pp 616-628, june 1975.

[FUL-74] : FULTON D.L.

A plasma-panel Interactive Graphic System.

Proceeding of the SID, 15-2, pp 74-80, 1974.

[GER-81] : GERRITSEN F., AARDEMA L.G.

Design and use of DIP-1: A fast flexible and dynamically microprogrammable image processor.

Pattern Recognition, vol 14, pp 319-330, 1981.

[GER-84] : GERRITSEN F., VERBECK P.W.

Implementation of cellular logic operators using 3x3 convolution and table lookup hardware.

Comput. Vision, Graphics, Image Processing, vol 27, pp 115-123, 1984.

[GOE-80] : GOETCHERIAN V.

From binary to grey tone image processing using fuzzy logic concepts.

Pattern Recogn., vol 12, pp 7-15, 1980.

[GOL-69] : GOLAY M.J.E.

Hexagonal parallel pattern transformations.

IEEE Trans. Comput., vol C-18, pp 733-740, 1969.

[GRA-80] : GRAHAM D., NORGRÉN P.E.

The Diff 3 analyzer : A parallel/serial Golay image processor.

Real Time Medical Image Processing. M. Onoe, K Preston and

A.Rosenfeld, Eds London. pp 163-182 Plenum 1980.

[HAD-57] : HADWIGER R.M.

Vorslesunger über Inhalt, Oberfläche und Isoperimetrie. Berlin, Springer, 1957.

[HAR-85] : HARALICK R.M.

A reconfigurable systolic network in computer vision.

IEEE Computer Society Workshop on Computer Architecture for  
Pattern Analysis and Image Database Management. Miami Beach, pp  
507-515, Nov 18-20, 1985.

[HAR-86] : HARALICK R.M., STERNBERG S.R., ZHUANG X.

Grayscale morphology.

Proc. IEEE Conf. Vis Pattern Recogn., Miami, FL, pp 543-550, 1986.

[HAR-87] : HARALICK R.M., STERNBERG S.R., ZHUANG X.

Image Analysis using Mathematical Morphology.

IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol  
PAMI-9, num 4, pp 532-550, July 1987.

[HEG-85] : HEGRON G.

Synthèse d'image : algorithmes élémentaires.

Dunod, 1985.

[HOL-84] : HOLLINGAM J.

Machine Vision.

IFS Pub., 113 pages, 1984.

[KIM-83] : KIM S.D.

Etude et développement de techniques de codage et de caractéristiques  
d'images vidéo.

Thèse de Docteur Ingénieur, INP Toulouse, 1983.

[KIM-85] : KIMMEL M.J., JAFFE R.S., MANDERVILLE J.R., LAVIN M.A.

MITE: Morphic image transform engine, an architecture for  
reconfigurable pipelines of neighborhood processors.

IEEE Computer Society Workshop on Computer Architecture for  
Pattern Analysis and Image Database Management. Miami Beach, FL,  
pp 493-500, Nov 18-20, 1985.

- [KIR-57] : KIRSCH R.A., CAHN L., RAY C., URBAN G.H.  
Experiments in processing pictorial information with digital computer.  
Proc. Eastern Joint Comput. Conf. pp 221-229, 1957.
- [KLE-72] : KLEIN J.C., SERRA J.  
The texture analyzer.  
J.Microscopy, vol 95, pt. 2, pp 349-356, Apr 1972.
- [KRU-77] : KRUSE B.  
Design and implementation of a picture processor.  
Science and Technology dissertation, Univ. Linköping, Linköping,  
Sweden, Rep 13, 1977.
- [KUN-87] : KUN A., MITRA S.  
A new algorithm of image edge extraction using a statistical classifier  
approach.  
P.A.M.I. Vol 9, 4, jul 1987.
- [LAT-84] : LATROUS S.  
Reconnaissance des formes et classification automatique du  
zooplancton par analyse d'images.  
Thèse de Docteur Ingénieur, Rennes, 1984.
- [LEE-86] : LEE J.S.J., HARALICK R.M., SHAPIRO L.G.  
Morphologic edge detection.  
I.E.E.E., pp 369-373, 1986.
- [LEL-79] : LELIEVRE J.L.  
Processus d'acquisition et de traitement d'images en temps réel.  
Application à l'inspection automatique.  
Thèse de Docteur Ingénieur, Lille, 1979.



[LEO-85] : LEONARD P.F.

Pipeline architectures for real-time machine vision.  
IEEE Computer Society Workshop on Computer Architecture for  
Pattern Analysis and Image Database Management. Miami Beach, FL,  
pp 502-505, Nov 18-20, 1985.

[LES-78] : LESTER J.M., WILLIAMS H.A., WEINTANB B.A., BRENNER J.F.

Two graph searching techniques for boundary finding in white blood  
cell images.  
Computer in Biology and Medecine, 8, pp 293-308, 1978.

[LEV-81] : LEVIALDI S.

Proc. NATO Advanded Study Institute.  
Reidel Publishing Company, p 105, 1981.

[LIE-78] : LIEBERMAN H.

How to color in a Coloring Book.  
Comp. Graphics, vol 12, Aug 1978.

[LIT-79] : LITTLE W.D.D., HEUFT R.

An area shading graphics display system.  
IEEE Trans. Comp., vol C-28, pp 528-531, 1979.

[MAN-85] : MANDEVILLE J.R.

Novel method for analysis of printed circuit images.  
IBM J. Res Develop., vol 29, pp 73-86, Jan. 1985.

[MAR-86] : MARAGOS P.

Tutorial on advances in morphological image processing and analysis.  
Proc. SPIE, Vis. Commun. Image Processing, vol 707, pp 64-74, 1986.

[MAR-85] : MARAGOS P., SCHAFFER R., W.

A unification of linear, median, order-statistics, and morphological filters under mathematical morphology.

IEEE Int. Conf. Acoustics, Speech and Signal Processing. Tampa. pp 1329-1332, Mar 1985.

[MAR-86] : MARAGOS P., SCHAFFER R., W.

Applications of morphological filtering to image processing and analysis.

IEEE Int. Conf. Acoustics, Speech and Signal Processing. Tokyo. pp 2067-2070, Apr 1986.

[MAR-86] : MARAGOS P., SCHAFFER R., W.

Morphological skeleton representation and coding of binary images .

IEEE Trans. on Acoustics, Speech and Signal Processing. Vol ASSP-34, pp 1228-1244, Oct 1986.

[MAR-87] : MARAGOS P., SCHAFFER R., W.

Morphological Filters-Part I : Their Set-Theoretic Analysis and Relations to linear Shift-Invariant Filters.

IEEE Trans. on Acoustics, Speech and Signal Processing. Vol ASSP-35, Num 8, pp 1153-1169, August 1987.

[MAR-87] : MARAGOS P., SCHAFFER R., W.

Morphological Filters-Part II : Their Relations to Median, Order-Statistic and Stack Filters.

IEEE Trans. on Acoustics, Speech and Signal Processing. Vol ASSP-35, Num 8, pp 1170-1184, August 1987.

[MAR-82] : MARTINEZ F.

Vers une approche systématique de la synthèse d'images - Aspect logiciel et matériel.

Thèse d'Etat, INPG, Grenoble, Nov. 1982.

[MAT-75] : MATHERON G.

Random Sets and Integral Geometry.  
Wiley, New York, 1975.

[MCC-85] : MCCUBBERY D.L., LOUGHEED R.M.

Morphological image analysis using a raster pipeline processor. IEEE  
Computer Society Workshop on Computer Architecture for Pattern  
Analysis and Image Database Management. Miami Beach, FL, pp 444-  
452, Nov 18-20, 1985.

[MER-73] : MERILL R.D.

Representation of contours and regions for efficient computer search.  
C.A.C. M. vol 16, n°2, pp 69-82, 1973.

[MEY-79] : MEYER F.

Iterative image transformations for an automatic screening of cervical  
smears.  
J. Histochem. Cytochem, vol 27, pp 128-135, 1979.

[MIC-78] : MICLET F., GRENIER H, KALISZEWSKI M.

Reconnaissance de formes.  
Cours E.N.S.T., 1978

[MIN-03] : MINKOWSKI H.

Volumen und oberfläche.  
Math. Ann. vol 57, pp 447-495, 1903.

[MON-76] : MONTELLI A.

An application of heuristic search methods to edges and contour  
detection.  
Commun. A.C.M. 19,2 pp 73-83, 1976

[MOO-68] : MOORE G.A.

Automatic scanning and computer processes for the quantitative analysis of micrographs and equivalent subjects.

Pictorial Pattern Recognition, G.C. Cheng et al, Eds. Washington DC: Thompson, pp 275-326, 1968.

[NAK-78] : NAKAGAWA Y., ROSENFELD A.

A note on the use of local min and max operations in digital picture processing.

IEEE Trans., Syst., Man., Cybern., Vol SMC-8, pp 632-635, 1978.

[NAL-86] : NALWA V., BINFORD T.

On detecting edges.

P.A.M.I. Vol 8, 6, nov 1986.

[NIL-71] : NILSON N.J.

Problem solving methods in artificial intelligence

Mac Graw Hill, N Y 1971

[NEW-73] : NEWMAN W. ET SPROULL R.

Principles of iterative computers graphics.

Mc Grawhill, N.Y., 1973

[PAV-78] : PAVLIDIS T.

Filling algorithms for raster graphics.

Dep. Elec. Eng. and Comp. Sci., Princeton Uni. NJ, Rep 238, jan 1978.

[PEL-84] : PELEG S., NAOR J., HARTLEY R., AVNIR D.

Multiple resolution texture analysis and classification.

IEEE Trans. Pattern Anal. Machine Intell. vol PAMI-6, pp 518-523, 1984.

[PEL-81] : PELEG S., ROSENDEL A.

A min max medial axis transformation.

IEEE Trans. Pattern Anal. Machine Intell. vol PAMI-3, pp 206-210,  
1981.

[POL-74] : POLKEY W. PILKEY O.

Mechanics of Solid.

Quantum Publisher, 441 pages, 1974.

[POS-87] : POSTAIRE J.G.

De l'image à la décision.

Dunod, 1987.

[PRA-78] : PRATT W.

Digital image processing.

Wiley Intersciences, New York, 1978.

[PRA-78] : PRATT W.

A pipeline architecture for image processing and analysis. IEEE

Computer Society Workshop on Computer Architecture for Pattern

Analysis and Image Database Management. Miami Beach, FL, pp 516-  
520, Nov 18-20, 1985.

[PRE-61] : PRESTON K.

Machine techniques for automatic identification of binucleate  
lymphocyte.

Proc. Fourth Int. Conf. Medical Electronics. Washington DC 1961.

[PRE-73] : PRESTON K.

Application of cellular automata to biomedical image processing.

Computer Techniques in Biomedecine and Medecine. Philadelphia.  
1873.

[PRE-66] : PREWITT J.M.S. ET MENDELSON M.

The analysis of cell images.

Ann. N.Y. Academic Sciences, Vol 128, pp 1035-1053, 1966.

[PRE-70] : PREWITT J.M.S.

Object enhancement and extraction in Picture Processing and Psychopictorics.

B.S.Lipkin and A.Rosenfeld, N.Y., Academic Press, 1970.

[RAA-78] : RAAMONT J.

Microprocessor applications of integer arithmetic.

Proc. IEEE, vol 66, pp 238-244, Feb 1978.

[RAK-79] : RAKHODAI I.

Etude et réalisation d'un extracteur rapide de caractéristiques d'images vidéo.

Thèse de Docteur Ingénieur, INP Toulouse, 1979

[RAM-75] : RAMER U.

Extraction of line structure from photographs of curved objects.

C.G.I.P. 4, 2, pp 81-103 1975.

[REY-75] : REYNOLDS C.W.

A multiprocessor approach to computer animation.

Undergrad Thesis, Dep. Elec. Eng. and Comp. Sci., Mass. Inst.

Technol., Cambridge, Aug 1975.

[ROD-83] : RODENACKER K., GAIS P., JUTTING U., BURGER G.

Mathematical morphology in grey images.

Proc. 1983 Euro. Signal Processing Conf.

[ROM-69] : ROMNEY G.W., WATKINS G.S., EVANS D.C.

Real-time Display of Computer Generated Half-tone Perspective Pictures.

Proceedings IFIP 1968, pp 973-978, Amsterdam, 1969.

[ROS-82] : ROSENFELD A., KAK A.C.

Digital picture processing.  
Academic Press. Vol 1 & 2. 1982.

[SAM-84] : SAMY R.

Traitement de séquence d'images télévision.  
Application à la poursuite de cibles.  
Thèse de l'Université. Aix Marseille. 1984.

[SKO-86] : SKOLNICK M.M.

Application of morphological transformations to the analysis of two  
dimensional electrophoretic gels of biological materials.  
Comput. Vis., Graph., Image Processing, vol 35, pp 306-332, 1986.

[SMI-79] : SMITH A.R.

Tint fill.  
Proc. Siggraph-ACM, pp 276-283, 1979

[SER-82] : SERRA J.

Stereology and structuring elements.  
J.Microscopy, pp 93-103, 1972.

[SER-82] : SERRA J.

Image analysis and mathematical morphology  
Academic Press. 1982

[SER-86] : SERRA J.

Introduction to Mathematical Morphology.  
Computer Vision, Graphics and Image Processing 35.  
pp 283-305, 1986.

[STE-79] : STERNBERG S.R.

Parallel architectures for image processing.  
Proc. IEEE Conf. Comput. Software Appl. Chicago, IL, pp 712-717,  
1979.

[STE-82] : STERNBERG S.R.

Esoteric iterative algorithms.

Proc. Second Int. Conf. Image Analysis and Processing. Italy. pp 15-18,  
1982.

[STE-83] : STERNBERG S.R.

Biomedical image processing.

IEEE Comput. Mag. pp 22-34, Jan 1983.

[STE-86] : STERNBERG S.R.

Grayscale morphology.

Comput. Vis., Graph., Image Processing, vol 35, pp 333-355, 1986.

[STE-82] : STERNBERG S.R., SKLANSKY J., BISCONTE J.C.

Cellular computers and biomedical image processing.

Biomedical Images and Computers. Springer Verlag, pp 294-319, 1982.

[STE-83] : STERNBERG S.R., STERNBERG E.S.

Industrial inspection by morphological virtual gauging.

IEEE Workshop Comput. Archit. Pattern Anal. Image Database  
Manag. Pasadena, Oct 1983.

[STU-80] : STUCK F.

Réalisation d'un système adaptatif de traitement d'images pour  
l'identification et la localisation de pièces en robotique.

Thèse de l'Université, Toulouse, 1980.

[SUZ-85] : SUZUKI S., ABE K.

New fusion operations for digitized binary images and their  
applications.

IEEE Trans. Pattern Anal. Machine Intell., vol PAMI-7, pp 638-651,  
Nov 1985.



[TOU-87] : TOUZANI A.

Classification automatique par détection des contours des modes des fonctions de densité de probabilité multivariées et étiquetage probabiliste.

Thèse d'Etat. Lille. 1987.

[UNG-58] : UNGER S.H.

A computer oriented to spatial problems.

Proc. IRE, vol 46, pp 1744-1750, 1958.

[VIE-88] : VIEREN C.

Segmentation de scènes dynamiques en temps réel. Application au traitement de séquence d'images pour la surveillance de carrefours routiers.

Thèse de Doctorat, Lille, 1988.

[VIS-87] : DOCUMENTATION VISIONIX.

Digital Design, Les Ulis, 1987.

[WAN-84] : WANG S.

Automatic multithresholding.

C.G.I.P., Vol 25, pp 6-67, 1984.

[WER-85] : WERMAN M., PELEG S.

Min-max operators in texture analysis.

IEEE Trans. Pattern Anal. Machine Intell. vol PAMI-7, pp 730-733, 1985.

[WES-78] : WESZKA J.S.

A survey of threshold selection techniques.

Computer graphics and image processing. Vol 7. pp 259-265. 1978.

[WIL-85] : WILSON S.

The Pixie-5000. A systolic array processor.

IEEE Computer Society Workshop on Computer Architecture for  
Pattern Analysis and Image Database Management. Miami Beach, FL,  
pp 477-483, Nov 18-20, 1985.

[WYL-67] : WYLIE C., ROMNEY G., EVANS D., ERDHAL A.

Half-tone Prespective Drawings by Computer.

Proceedings fall joint Computer Conference, Thompson Books,  
Anaheim, 1967.

