

50376
1979
226

~~EXCLU DU PRET~~

50376
4979
226

N° d'ordre : 231

THÈSE

présentée à

L'UNIVERSITÉ DES SCIENCES ET TECHNIQUES DE LILLE

pour l'obtention du titre de

DOCTEUR - INGÉNIEUR

par

Jean-Luc LELIEVRE

Ingénieur ISEN

**PROCESSUS D'ACQUISITION ET DE TRAITEMENT
D'IMAGES EN TEMPS REEL :
APPLICATION A L'INSPECTION AUTOMATIQUE
D'UN MATERIAU TEXTILE.**



Soutenue le 14 juin 1979, devant la Commission d'Examen
Membres du Jury : MM.

P. VIDAL

Président

M. BOURTON

Rapporteur

M. NAJIM

Examineur

J.M. TOULOTTE

Examineur

G. MAZINGUE

Invité

A V A N T - P R O P O S

Le travail présenté dans ce mémoire a été effectué au Centre d'Automatique de l'Université des Sciences et Techniques de Lille 1 et à l'Institut Textile de France.

Nous tenons à remercier Monsieur le Professeur Pierre VIDAL qui nous a accueilli au sein de son Laboratoire. Qu'il trouve ici l'expression de notre gratitude pour l'honneur qu'il nous fait en présidant notre jury.

Nous adressons notre plus grande reconnaissance à Monsieur MAZINGUE, Directeur de l'I.T.F. Nord pour l'excellent accueil qu'il nous a réservé au sein de son Laboratoire. Qu'il soit remercié pour l'intérêt qu'il a sans cesse porté à ce travail et pour sa présence dans notre jury.

Nous sommes très reconnaissants et remercions vivement Monsieur le Professeur M.NAJIM, du Laboratoire d'Automatique de la Faculté des Sciences de Rabat (Maroc), pour sa présence dans notre jury. Qu'il soit également remercié pour l'accueil qu'il nous a réservé dans son Laboratoire, pour ses conseils et critiques constructifs lors de la rédaction de ce mémoire.

Nous exprimons ici notre reconnaissance à Monsieur TOULOTTE, Maître de Conférences à l'Université de Lille 1, qui a bien voulu s'intéresser à nos travaux et participer à notre commission d'examen.

Nous sommes très heureux de pouvoir exprimer ici toute notre gratitude à Monsieur Michel BOURTON, Maître de Conférences à l'Université de Valenciennes, qui nous a suivi et guidé durant toute cette étude. Qu'il en soit vivement remercié.

Notre travail n'a pu être mené à bien que grâce à la coopération et à l'aide du personnel de l'I.T.F. Nord, et en particulier grâce aux services de Monsieur OUTREMAN.

Que soient remerciés tous les membres du Laboratoire d'Automatique de Lille 1, qui ont contribué au maintien d'une ambiance chaleureuse et amicale.

Qu'il nous soit enfin permis de remercier les membres du Laboratoire d'Automatique (L.E.E.S.A.) de la Faculté des Sciences de Rabat pour leur amical accueil et pour leur contribution à la mise en forme de ce mémoire.

INTRODUCTION GENERALE

Les travaux exposés dans ce mémoire s'intègrent dans le domaine de la reconnaissance de formes, et traitent plus particulièrement d'une procédure originale d'acquisition et d'identification d'objets, par le biais d'images digitales. Cette recherche a eu pour origine un problème industriel de détection de défauts en temps réel dans une matière textile, et a abouti à la réalisation d'un prototype.

Ce mémoire se décompose en deux ensembles principaux :

Une première partie, sous-divisée en quatre chapitres, définit les procédures qui permettent, à partir d'images, d'aboutir à la reconnaissance de formes, et ce dans le cas général dans lequel est intégrée notre application.

- Le premier chapitre rappelle les domaines d'application du traitement d'images et résume les méthodes généralement employées, pour mettre en évidence finalement le créneau des applications temps réel .

- Un second chapitre est consacré à la procédure de détection des objets, toujours dans une optique temps réel.

- Le troisième chapitre, étape intermédiaire entre la détection et la reconnaissance permet de définir les objets, non plus par leur représentation imagée initiale, mais par leur contour.

- Le quatrième et dernier chapitre de cette partie est beaucoup plus consacré à un exposé des méthodes de calcul de paramètres utiles à l'identification des formes , qu'à la méthode de reconnaissance elle-même simplifiée, dans notre cas par la présence d'un nombre réduit de classes.

La seconde partie décrit le prototype en deux étapes :

- Le cinquième chapitre est consacré à la définition de l'interface entre une caméra à base de photodiodes et un microcalculateur, éléments dont le faible coût faciliteront le développement industriel du prototype.

- Le sixième chapitre aborde les problèmes logiciels rencontrés lors de la transposition des procédures définies dans la première partie.

- Le septième et dernier chapitre présente les performances actuelles du prototype et énonce quelques perspectives d'amélioration.

CHAPITRE I

- SYSTEME DE RECONNAISSANCE DE FORMES.

La reconnaissance des formes est un processus naturel auquel l'homme est continuellement confronté dans la vie courante ; c'est ainsi que nous reconnaissons un objet parmi un univers d'objets de plus en plus nombreux et complexes. Même pour des opérations apparemment simples telles que la lecture et l'identification des sons, nous faisons appel à nos capacités de reconnaissance.

Chaque objet, mot et son est perçu par le biais de paramètres pertinents qui permettront à une situation nouvelle d'être associée à une situation déjà rencontrée. Ainsi par formalismes successifs nous tentons d'approcher les mécanismes de l'intelligence humaine, sans prétendre à une explication complète du fonctionnement du cerveau. L'intelligence artificielle met en oeuvre des processus qui, grâce à l'ordinateur, permettent de poser le même type de problèmes qu'envisage le cerveau humain; la reconnaissance des formes n'est qu'une partie de ce vaste ensemble mais en est sans doute la partie la plus développée.

I.1. Domaines d'applications de la reconnaissance de formes

Les progrès actuels dans la connaissance des mécanismes humains de reconnaissance, les récents perfectionnements techniques, tant au niveau des capteurs, qu'au niveau des ordinateurs, nous permettent de croire que l'implantation de machines automatiques de reconnaissance n'en est qu'à ses premiers balbutiements. Les domaines de recherche sont multiples, comme le montre le tableau non exhaustif de la figure I.1.

TACHES	DONNEES	RESULTATS
Reconnaissance de l'écriture (1,2,3,4)	signal optique	identification des caractères, mots, adresses postales...
Reconnaissance de la parole (5)	signal acoustique	Reconnaissance des phénomènes ; mots, locuteurs
Cartographie (6,7)	signal optique	Prévision du temps Etudes des ressources terrestres...
Domaine biomédical (8,9)	- signaux E.E.G. - radios - données sanguines - signal optique	Comptage de cellules Identification des chromosomes, diagnostic
Robotique (10,11)	signal optique	Placement des pièces. Nettoyage du sol par robot....
Diverses applications (12,13)	signal optique	Empreintes digitales reconnaissance du profil d'un visage.

Figure I.1. : Domaines d'application de la reconnaissance des formes.

I.2. Définition d'un système de reconnaissance

Schématiquement, un système de reconnaissance est un dispositif qui se scinde en quatre organes :

- Perception.

Le premier organe, constitué d'un ou plusieurs capteurs, permet la perception des formes. Il faut noter l'importance du choix du capteur, car il conditionne le fonctionnement de l'ensemble du système, tant au point de vue qualité que rapidité d'acquisition.

- Traitement

La seconde étape sur laquelle nous reviendrons plus longuement dans la suite de ce chapitre permet d'extraire la forme de son environnement (extraction de contour, par exemple). Cette étape est bien souvent précédée d'un traitement visant à améliorer la qualité de l'information en sortie du capteur.

- Paramétrisation

Il s'agit là d'un des problèmes les plus importants de la reconnaissance des formes : celui du choix des paramètres les plus pertinents au sens de la reconnaissance.

- Décision

La dernière étape est celle de la décision qui s'opère grâce à un algorithme de comparaison de la forme acquise à des formes de référence.

Les méthodes de reconnaissance sont très nombreuses, elles peuvent être classées en deux catégories : les méthodes syntaxiques (14) où chaque forme est décrite en fonction de ses composantes et de leurs interrelations. Suivant le mode de description des formes, on distingue encore les méthodes déterministes et probabilistes (15)(16) dont la procédure de classement est obtenue par apprentissage, ce qui permet de construire des "fonctions de discrimination" par approches successives.

I.3. Traitement d'images

I.3.1. Définition d'une image digitale

Nous traitons plus particulièrement dans ce mémoire le cas d'un système de reconnaissance où la forme est décrite par son image optique.

Si l'on considère une image issue, par exemple, d'une photographie que nous voulons analyser à l'aide d'un ordinateur, la première étape consiste à échantillonner le plan image. Celui-ci est décomposé en petits éléments de surface de dimension $a \times a$, où a est le pas d'échantillonnage. L'image est alors représentée par une matrice dont chaque élément $T(i,j)$ (appelé pixel : dérivé de Picture Element) est proportionnel à l'intensité lumineuse émise par une surface élémentaire.

I.3.2. Techniques de traitement d'images.

Le domaine d'application du traitement d'images est tellement vaste et diversifié qu'aucune théorie générale ne peut être dégagée; toutefois, il est possible de tirer quelques idées directrices.

Les méthodes sont présentées dans l'ordre chronologique dans lequel elles se déroulent.

I.3.2.1. Compression et codage d'images (17,18)

Cette étape du traitement s'impose particulièrement lors de la saisie d'images pour des applications spatiales (clichés photographiques pris par satellite), pour lesquelles le problème crucial est la réduction de bande passante de transmission.

Une image, même si elle paraît complexe (paysage), est un ensemble redondant d'informations. Il est possible de réduire très fortement la quantité de données de l'image, sans en altérer, outre mesure, la qualité.

On peut distinguer deux types de procédés :

- Traitement local : ces méthodes, se basant sur le fait que la valeur d'un "pixel " peut être prédite par les valeurs des pixels voisins, font appel aux techniques de codage de l'information.

- Traitement global : ces méthodes nécessitent, au préalable, la représentation de l'image dans le plan des fréquences spatiales obtenues par des transformations orthogonales telles que transformées de FOURIER, WALSH-HADAMART. La compression consiste à réduire la quantification de certaines parties du spectre fréquentiel dont le contenu informationnel, au sens optique, est très faible.

Il ne faut pas oublier de citer la compression d'images couleur, (19,20) obtenues à partir de données multispectrales.

I.3.2.2. Restauration d'images

Le but de la restauration d'images est d'éliminer certaines imperfections inhérentes au mécanisme de formation des images : aberrations optiques, bruit électronique etc.... Parmi les très nombreuses techniques de restauration, nous pouvons citer l'utilisation de filtres digitaux linéaires, lorsque nous pouvons modéliser la dégradation par une transformation linéaire (21,22).

Des traitements locaux non linéaires (l'image est traitée par parties) permettent de rehausser les contrastes et donc d'améliorer la netteté (23).

Un autre domaine de la restauration d'images concerne les images tri-dimensionnelles $T(i,j,k)$ reconstituées à partir d'images obtenues par des capteurs mono et bi-dimensionnels (24,25).

I.3.2.3. Segmentation d'image, description des formes (26,27,28,29)

Cette partie du traitement a pour but d'extraire la forme ou une partie d'image de son environnement, parfois très complexe. Des régions d'une image peuvent se distinguer par des niveaux de gris différents de ceux du contexte. Parfois, les régions se caractérisent par des textures différentes : par exemple, une chevelure présente une "granulation", au sens optique, différente de celle du visage.

La détection de régions, dans le cas où celles-ci ont des niveaux de gris caractéristiques, peut s'obtenir par seuillage de l'image. Lorsque ce seuil n'est pas connu, des techniques permettent de le calculer grâce à une analyse statistique des distributions de niveaux de gris.

Une partie du chapitre III est consacrée aux méthodes de recherche de contour qui permettent de décrire une région par l'ensemble de points de la ligne frontière entre la région et son environnement.

I.4. Matériel utilisé pour le traitement d'images.

I.4.1. Utilisation des méthodes de traitement d'images en temps différé

Le rapide exposé des méthodes de traitement d'images effectué dans le sous-chapitre précédent nous amène à faire quelques constatations pour le choix du matériel informatique et périphérique.

De par la complexité du problème ainsi posé, les méthodes requièrent généralement des temps de calcul très longs et ne peuvent être exécutées au fur et à mesure de l'acquisition. En effet, la qualité de prise de vue dépend de la rapidité d'acquisition de l'image numérique. Par ailleurs certaines techniques de traitement nécessitent la connaissance de l'image complète avant d'entamer le traitement. En conséquence, le processus

de reconnaissance se décompose en deux étapes :

- acquisition et rangement de l'image en mémoire de masse (disque, par exemple). Cette opération peut être réalisée par l'intermédiaire du calculateur, ou, si l'acquisition est trop rapide, directement du capteur vers la mémoire grâce à un processeur spécialisé.

- lecture en mémoire et traitement de parties d'images à un rythme plus lent. Dans le cas où la taille de l'image est petite, et celle de la mémoire centrale importante, la mémoire de masse n'est pas nécessaire.

Evidemment, cette contrainte restreint le champ d'applications du traitement d'image. En effet, certaines applications requièrent un traitement en temps réel de façon à avertir un opérateur de la détection ou reconnaissance d'une forme (défaut, par exemple), ou à intervenir automatiquement sur une chaîne de fabrication.

I.4.2. Systèmes interactifs de traitement d'images (30,31,32)

Dans un but de recherche, certains laboratoires spécialisés en traitement d'images, ont constitué un matériel le plus "universel" possible. L'acquisition d'images peut se réaliser par plusieurs capteurs : analyseur par flying spot dans le cas d'une définition d'image très grande, caméras de télévision ou réseaux de photodiodes lorsque la définition requise est moyenne. De tels systèmes permettent un traitement en conversationnel, de telle façon que l'utilisateur puisse accéder à une bibliothèque de programmes usuels de traitement d'images et aussi constituer sa propre bibliothèque d'images et de programmes spécifiques au problème posé. Certains laboratoires ont élaboré, dans ce but, des langages spécifiques au traitement d'images, formés de macrocommandes.

Pour que le système soit inter-actif, c'est-à-dire pour que l'utilisateur ait la possibilité de réagir en cours de traitement et de changer éventuellement certains paramètres de programmes, un synthétiseur d'image devient nécessaire.

Celui-ci permet de reconstituer, sur une console de visualisation, l'image résultante d'un traitement et même pour certains systèmes, l'image en cours de traitement.

I.4.3. Notion de temps réel en traitement d'images

Dans le domaine industriel, lorsque les traitements d'images en temps réel ne sont pas impératifs, nous retrouvons le même type de matériel défini précédemment :

- calculateurs puissants possédant une mémoire centrale importante
- périphériques constitués de mémoires de masse à accès rapide.
- consoles de visualisation pour le contrôle des résultats.

Des recherches sont toujours en cours en vue de la reconnaissance en temps réel de caractères alphanumériques imprimés ou même manuscrits. Pour de telles applications, un traitement hybride peut être utilisé :

- traitements, en partie, optiques qui ont la propriété d'être très rapides.
- utilisation d'opérateurs analogiques et numériques cablés.
- traitement logiciel lorsque l'analyse de l'image devient complexe.

Certaines réalisations industrielles s'effectuent en temps réel : dispositif automatique de contrôle de circuits intégrés (33), de positionnement d'objets (34). La plupart de ces applications requiert une recherche de contour de forme en temps réel, opération simplifiée par le fait que la composition des images est simple et que le contexte ou support des formes est de nature homogène.

I.5. Conclusions

La dernière partie de ce chapitre met en évidence qu'un créneau d'applications est très peu exploité : il s'agit de tous les problèmes d'inspections, jusqu'à maintenant visuelles, de matières telles que papier, tissu dans

le but de comptabiliser des formes (défauts, objets). Dans la majorité des cas, les dimensions importantes de la matière analysée nécessitent un déroulement continu devant un opérateur, par conséquent, la détection des formes doit s'effectuer en temps réel.

Lorsque l'on veut connaître, non seulement, la position des objets, mais aussi leur nature, une représentation imagée des formes devient nécessaire. Cette tâche devient plus délicate, en temps réel, lorsque les images sont fortement bruitées, ce qui est le cas dans notre application.

Sans prétendre trouver une solution à tous les problèmes de ce type, nous définissons, dans les prochains chapitres, les procédures qui conduisent de la détection des objets **ou** formes à leur identification.

C H A P I T R E I I

- ACQUISITION DES SOUS-IMAGES EN TEMPS REEL.

Au delà des impératifs d'acquisition d'image abordés dans les premiers paragraphes, l'essentiel de ce chapitre est consacré aux procédures d'acquisition, en temps réel, de formes dans une image très "bruitée". Pour réduire au maximum les zones acquises et donc les temps de traitement ultérieurs, l'algorithme développe une procédure double :

- détection des zones de présence de formes*
- traitement effectué sur ces zones pour confirmer ou infirmer cette présomption de présence de forme.*

Enfin, pour faciliter le traitement ultérieur de recherche de contour (chapitre III), les formes sont acquises par cadrage minimal sans réduction essentielle d'informations sur leur structure

II.1. Présentation des images

II.1.1. Traitement d'image en temps réel

Avant de présenter les procédures de recherche de formes dans une image, il est indispensable de situer les domaines d'application et les limites de la méthode.

Au delà du problème textile qui en est le support d'application, la méthode s'adresse essentiellement à la détection de formes dont les dimensions sont très faibles par rapport aux dimensions de l'image complète. Les procédures mises en œuvre s'appliquent tout particulièrement dans le cas où les dimensions "m" et "n" de la matrice $T(i,j)$ sont telles que "m" peut être considéré comme infini devant n.

Ce cas est rencontré lors de l'analyse optique d'un support imagé dont la longueur est très grande. Pour ce type d'applications, l'acquisition de l'image est généralement réalisée par un capteur à structure linéaire (ligne de photodiodes); la deuxième dimension de l'image est fournie par le déplacement soit de la matière analysée, soit du capteur lui-même .

Dans un tel cas, la vitesse (v) du support ou du capteur, le pas d'échantillonnage (a) de l'image {choisi en fonction de la plus petite taille de forme à détecter} et la fréquence F d'acquisition des pixels T (i,j) sont liés par la relation :

$$F = \frac{v \times n}{a} \quad (n : \text{nombre de pixels sur une ligne})$$

Dans notre application, qui est l'objet de la deuxième partie de ce mémoire, la fréquence d'acquisition (F) est de l'ordre de 100.000 pixels par seconde . Il est évident qu'un traitement logiciel de toutes les informations de l'image, en temps réel, est impossible.

Une solution à ce problème consiste à ranger l'image dans une mémoire de masse, et à traiter ultérieurement les informations à une cadence plus faible. Dans notre cas, l'utilisation d'une mémoire de masse n'est pas concevable, car le débit élevé d'informations ainsi que la taille de cette mémoire nécessitent un matériel trop coûteux pour l'application envisagée.

Par conséquent, la détection et le traitement doivent être quasi-simultanés à l'acquisition. Cet impératif de traitement en temps réel nous a conduit à développer des méthodes adaptées au mode d'acquisition, c'est à dire, ligne par ligne.

II.1.2. Conditions d'analyse

L'analyse de l'image, définie par les paragraphes suivants, est subordonnée aux contraintes qu'imposent les formes à détecter et leur environnement.

Le conditionnement de la méthode est lié aux aspects suivants :

- fluctuations en largeur et position du support des formes.
- variations de luminosité du support.
- nature des formes.

a) fluctuations du support

L'existence de fluctuations en largeur et en position du support imagé peut provoquer la non-détection de certaines formes par perte d'informations en limite de zone analysée ; la largeur analysée ($n \times a$) doit être légèrement supérieure à la largeur du support (figure II.1)

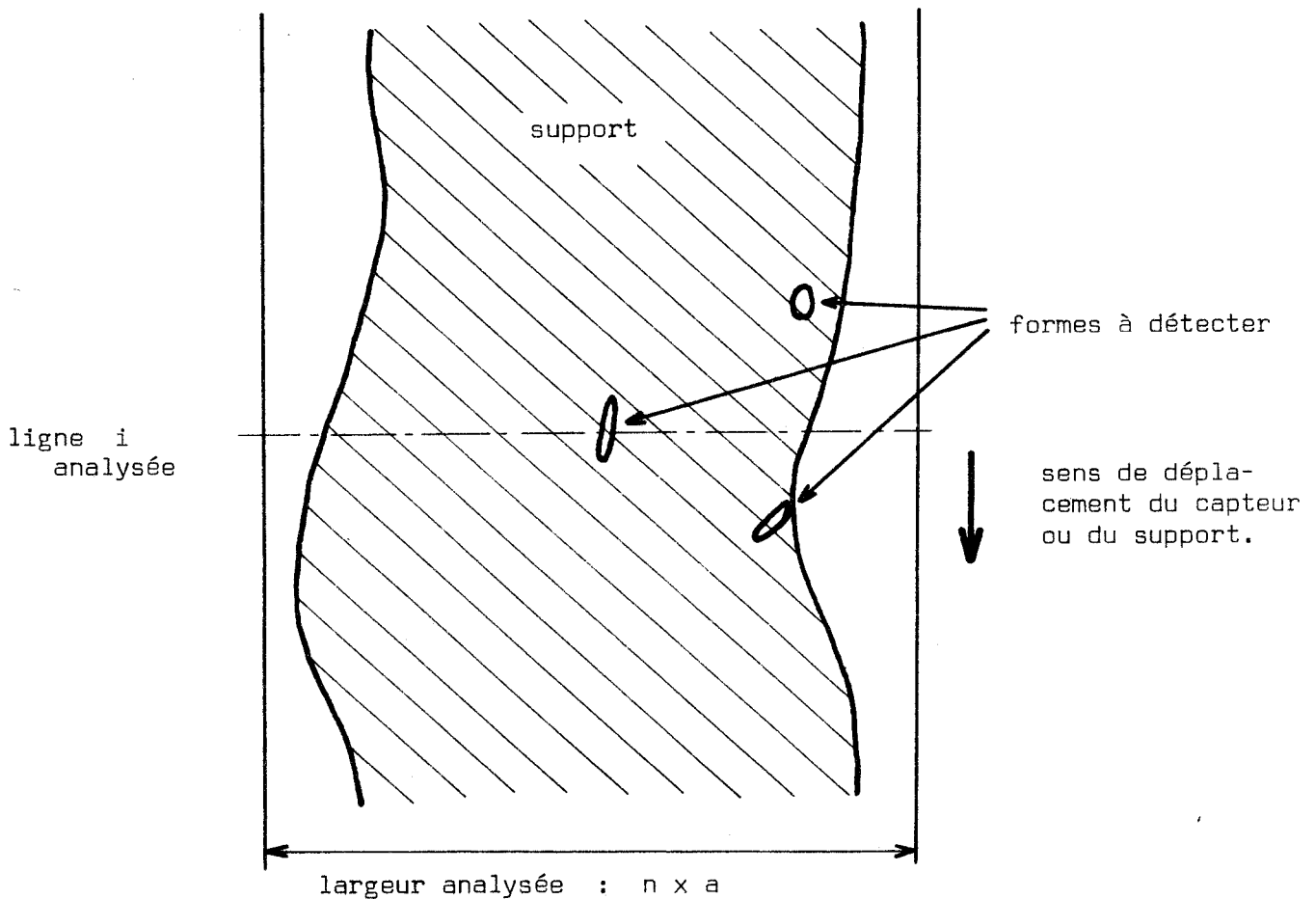


Figure II.1. : Présentation du support

b) variations lumineuses du contexte

Hormis les "bruits" inhérents à tout système d'acquisition opto-électronique, la nature hétérogène du support (exemple : trame d'un tissu, texture d'un papier, fibres de laine) peut être considérée comme source d'un "bruit de fond" dans l'image.

A cet aspect "granulaire" de l'image du support, vient s'ajouter un autre phénomène : le support peut présenter une densité optique moyenne différente d'une région à l'autre de l'image. Cette fluctuation de luminosité est supposée néanmoins faible devant les contrastes des formes, de telle sorte que l'on puisse considérer qu'une forme se présente dans un contexte à densité optique pratiquement constante.

Les méthodes décrites dans ce chapitre ont été spécialement adaptées aux images pour lesquelles le contexte ou image du support est fortement "bruité"

c) nature des formes

Pour simplifier et sans pour autant limiter les performances de la méthode, certaines contraintes aux formes sont imposées :

- chaque forme ne présente qu'une seule coupe par ligne. Les formes ne sont donc définies que par une frontière gauche et une frontière droite sur une ligne d'indice i . Une partie du chapitre VII est consacrée à un essai de généralisation de la méthode. Dans le cas général, plusieurs coupes peuvent être présentes sur une ligne d'indice i .

- la taille de cette coupe, que l'on peut appeler largeur de la forme est supposée inférieure à une largeur prédéterminée (l_{max}). Cette largeur maximale est fixée par la nature des formes à détecter.

II.2. Définition des segments de forme

Le but de ce sous-chapitre est de présenter une procédure d'analyse par parties d'images contenant les formes, appelées sous images, en vue d'établir un mode d'acquisition et de traitement adapté à la nature séquentielle de l'analyse optique. Pour cela nous allons examiner la coupe

d'un défaut obtenu sur une ligne d'indice i .

Il convient de préciser que les exemples de ce chapitre ont été obtenus à partir d'une analyse optique par transparence. Par conséquent, les valeurs faibles $T(i,j)$ correspondent à des zones opaques de l'image.

II.2.1. Définition des sous-images dans le cas idéal

Le cas idéal d'une forme, représenté sur la figure II.2 est caractérisé par les conditions suivantes :

- le contexte, ou image du support, présente une intensité lumineuse homogène. Cette régularité se traduit au niveau de la représentation imagée par les valeurs $T(i,j) = 1$

- l'intensité lumineuse émise par l'intérieur de la forme est également homogène ($T(i,j) = 0$).

- le contraste forme-contexte est supposé parfait, ce qui se traduit par l'absence des valeurs $T(i,j)$ comprises entre 0 et 1.

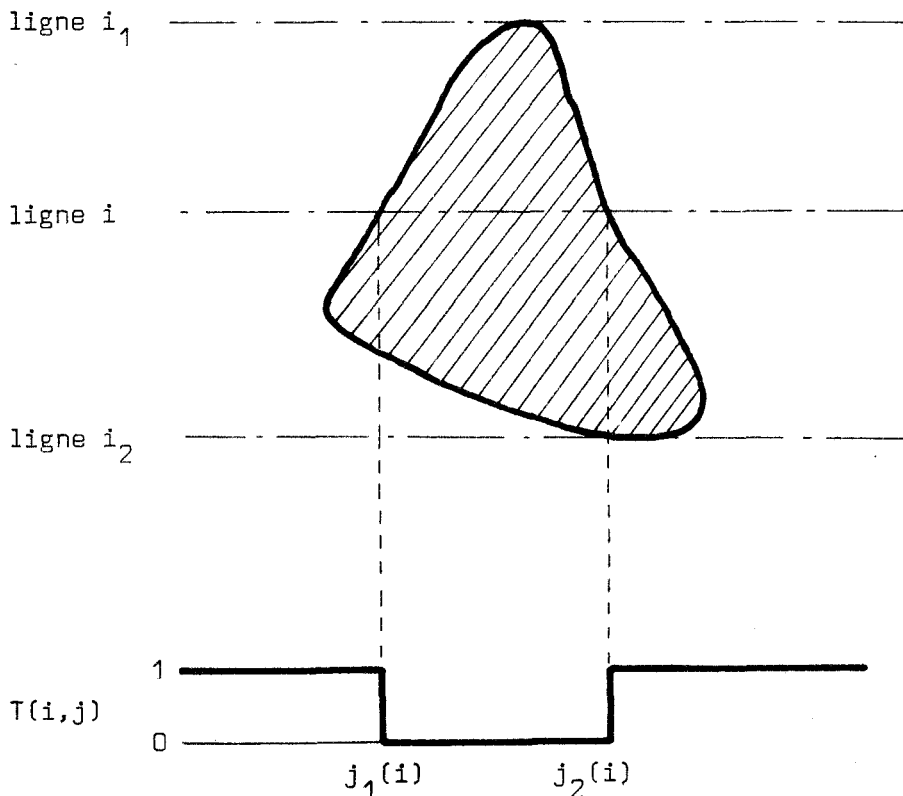


Figure II.2 :
Sous image dans le
cas idéal.

Sur une ligne d'indice i , on peut caractériser la forme par un triplet $s(i)$ appelé segment $\{i, j_1(i), j_2(i)\}$ où j_1, j_2 sont définis par :

$$j_1(i) = \inf (j) \text{ tel que } T(i,j) = 0$$

$$j_2(i) = \sup (j) \text{ tel que } T(i,j) = 0$$

La représentation de la sous-image est alors ramenée à l'ensemble des segments $s(i)$ dans un intervalle (i_1, i_2) où i_1 et i_2 sont respectivement les positions des lignes supérieure et inférieure encadrant la forme.

Cette définition n'est pas suffisante lorsque plusieurs formes sont présentes dans l'image . Il faut alors introduire une relation qui permette d'associer les segments $s(i)$ d'une même forme.

Deux segments $s(i) = \{ i, j_1(i), j_2(i) \}$ et

$s(i+1) = \{ i+1, j_1(i+1), j_2(i+1) \}$ sont dits connectés si les intervalles (j_1, j_2) ont une partie commune soit :

$$(j_1(i), j_2(i)) \cap (j_1(i+1), j_2(i+1)) \neq \emptyset$$

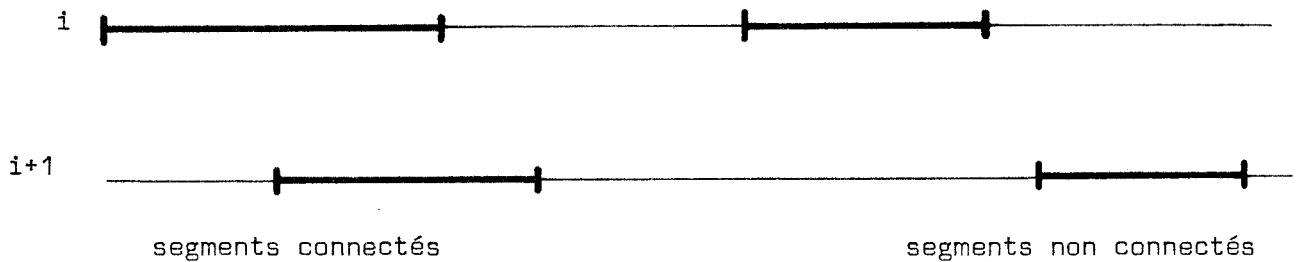


Figure II.3. : Connexion des segments.

II.2.2. Définition des segments de forme dans un cas intermédiaire

Pour préciser progressivement le processus utilisé dans le cas réel, considérons le cas intermédiaire où le contexte (F^*) et la forme (F) sont à niveaux de gris constants, mais où le contraste est imparfait (partie Z_c).

Ce cas est schématisé sur la figure II.4 où les courbes représentent les points iso-niveaux.

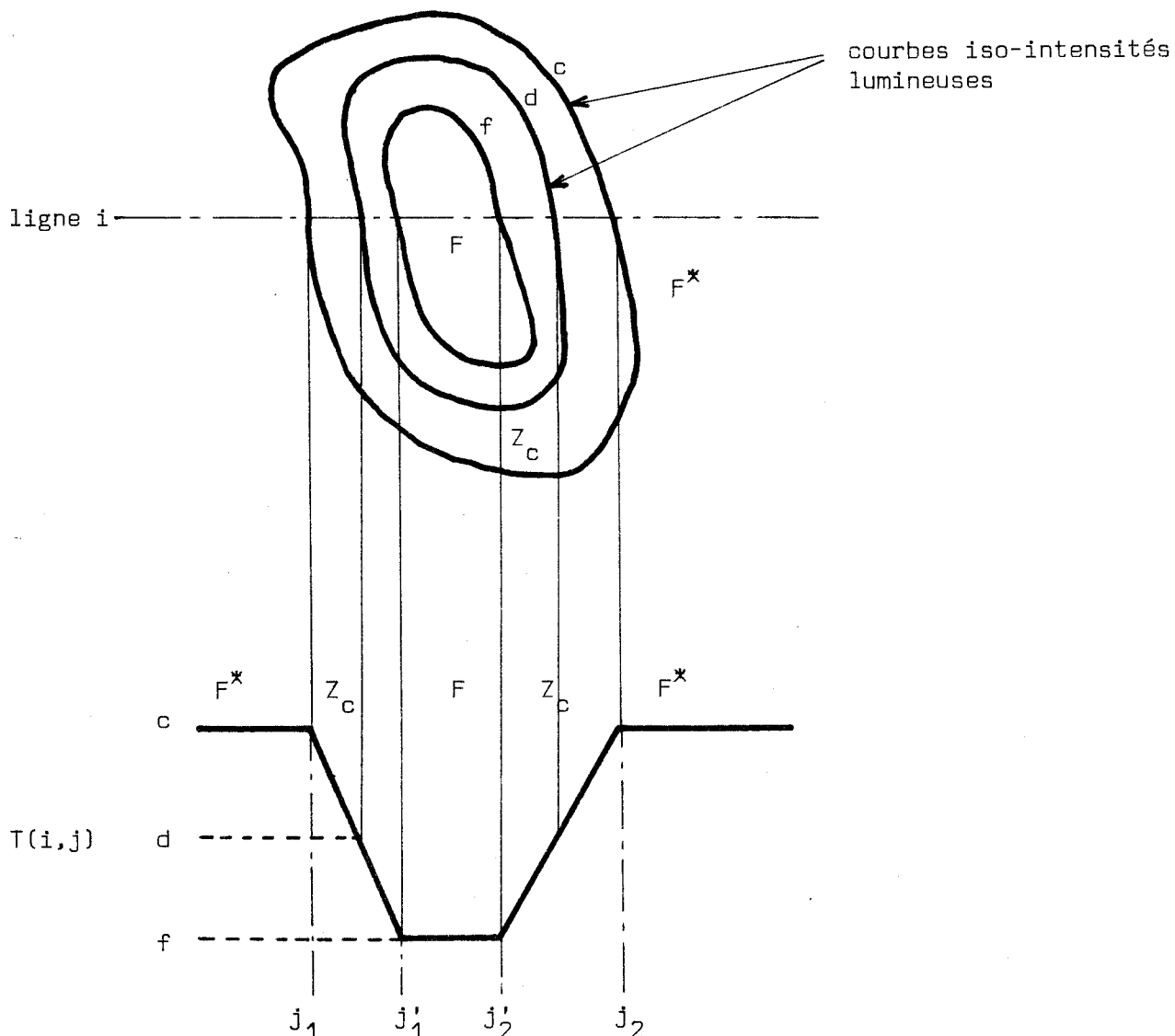


Figure II.4. : Sous-image d'une forme dans le cas intermédiaire

Une représentation binaire de l'image ($T(i,j) = 0,1$) n'est plus suffisante pour définir correctement la forme; une représentation multiniveau est alors nécessaire.

Comme on le constate sur la figure II.4, le segment $s(i)$ ne peut plus être défini simplement par un triplet. En effet, les frontières gauche et droite sont désormais caractérisées respectivement par les couples (j_1, j'_1) et (j_2, j'_2) tels que :

$$j'_1 = \inf j, \text{ tel que } T(i,j) = f$$

$$j'_2 = \sup j, \text{ tel que } T(i,j) = f$$

$$j_1 = \sup j, \text{ tel que } T(i,j) = c \text{ et } j < j'_1$$

$$j_2 = \inf j, \text{ tel que } T(i,j) = c \text{ et } j > j'_2$$

(Le segment $s(i)$ est alors défini par un quintuplet :

$$\{ i, j_1, j'_1, j_2, j'_2 \}$$

II.2.3. Définition des segments de forme dans le cas général

Le cas réel se définit par les conditions suivantes :

- le contexte (F^*) est de nature hétérogène (fibres de laine, trame d'un tissu ...) ce qui se traduit, au niveau de l'image $T(i,j)$, par la présence d'un "bruit" important. D'autre part, une forme peut se présenter dans une zone du contexte d'intensité $T(i,j)$ variable. Pour une même forme, les contextes droit et gauche sont légèrement différents.
- l'intensité lumineuse à l'intérieur de la forme peut être également faiblement variable.

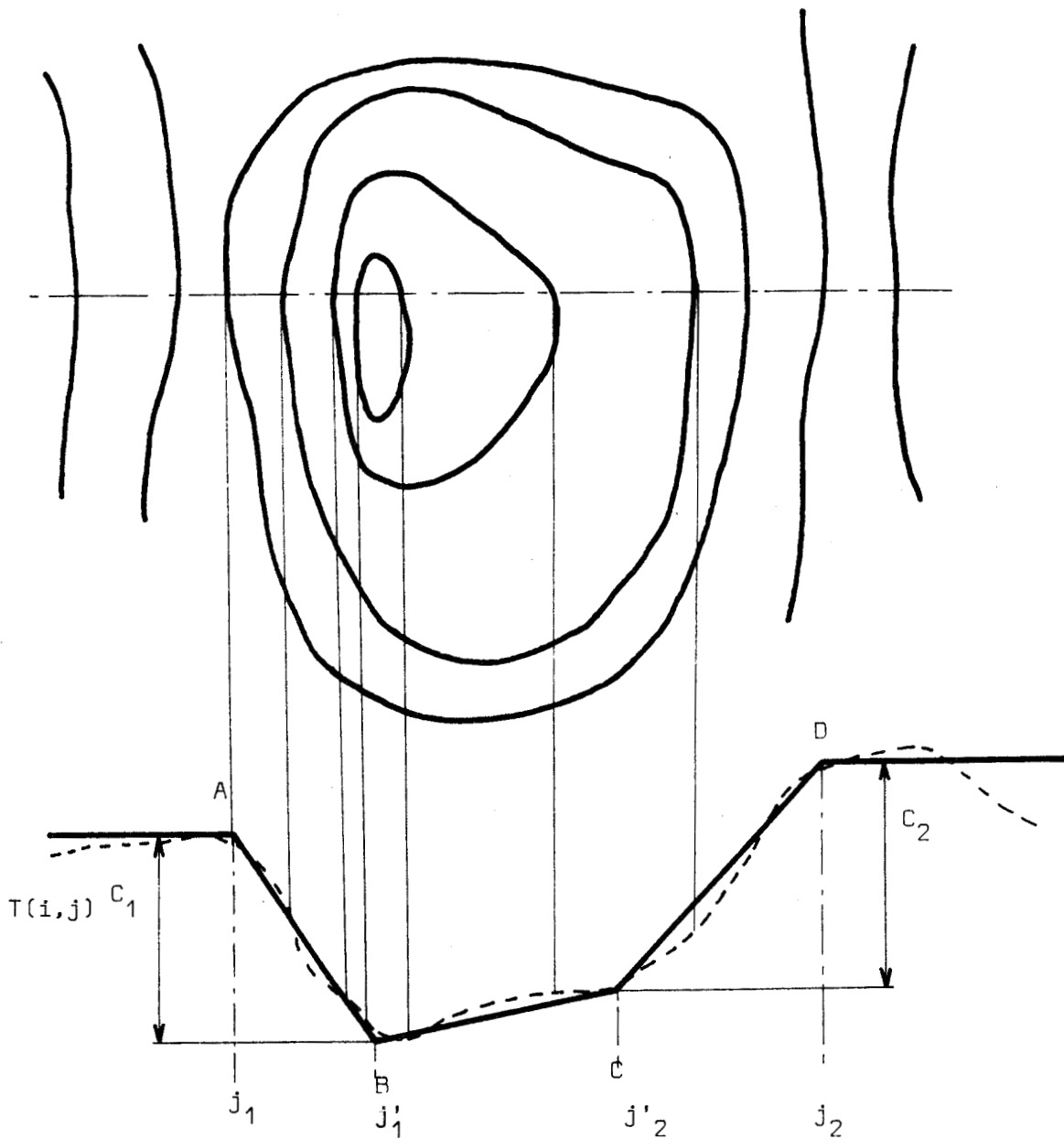


Figure II.5. : Sous-image d'une forme dans le cas réel

La présence du bruit dans le signal $T(i,j)$ ne nous permet pas de définir simplement les positions respectives j_1, j'_1, j_2, j'_2 des points A, B, C, D. On peut remarquer, néanmoins, que chacune de ces positions délimitent les zones caractéristiques du segment :

- pour des valeurs $j < j_1$
 - $j > j_2$
 - $j \in (j'_1, j'_2)$
- } le signal $T(i,j)$ présente des faibles variations

- pour les valeurs $j \in (j_1, j'_1)$ } les variations de $T(i, j)$
 $j \in (j'_2, j_2)$ } sont relativement élevées

Pour éviter de confondre des segments de forme avec des zones du contexte ayant la même allure qu'un segment, mais des contrastes moins prononcés, nous définissons $s(i)$ par un quintuplet: (i, p_1, p_2, c_1, c_2)

* p_1, p_2 sont les pentes gauche et droite représentatives des zones contrastées.

$$p_1 = \frac{T(i, j_1) - T(i, j'_1)}{j'_1 - j_1}$$
$$p_2 = \frac{T(i, j_2) - T(i, j'_2)}{j_2 - j'_2}$$

* c_1, c_2 sont appelés contrastes par analogie avec le contraste visuel

$$c_1 = T(i, j_1) - T(i, j'_1) ; c_2 = T(i, j_2) - T(i, j'_2)$$

On se rapproche du cas idéal précédemment étudié lorsque les pentes p_1, p_2 sont importantes, et lorsque les contrastes c_1, c_2 atteignent des valeurs suffisantes .

La connexité des segments, permettant d'associer les segments d'une même forme , est dans le cas réel , plus délicate à définir que dans le cas idéal. En effet, les frontières ne sont plus définies par un couple de points j_1, j_2 mais par des zones frontières (j_1, j'_1) et (j_2, j'_2) . Cette notion de connexité étant fortement liée à la procédure d'acquisition, sera étudiée ultérieurement (§ II.4.1)

II.3. Acquisition des sous images en temps réel

II.3.1. Principe de détection des formes

Le principe de localisation des formes consiste à rechercher dans l'image $T(i,j)$, au fur et à mesure de l'acquisition, des segments $s(i)$ représentatifs de formes.

Un traitement informatique de l'image complète, si simple soit-il, suppose une acquisition lente. En effet, la présence du bruit dans l'image complique la localisation des formes par voie logicielle, et par conséquent demande des temps de traitement importants.

Pour permettre une détection de formes en temps réel, la procédure se décompose en deux phases :

- prélocalisation par voie "matérielle" des sous-images.

Cette prélocalisation a pour but de repérer les positions de l'image (i, j_d^i) , pour lesquelles il y a de fortes présomptions de présence de forme.

- validation par voie "logicielle" des prélocalisations

Dans un domaine de l'image entourant une position j_d^i un traitement logiciel permet, dans un premier temps, de s'assurer de l'existence des points A, B, C, D définis précédemment. La connaissance de ces points caractéristiques permet ensuite le calcul des pentes (p_1, p_2) et des contrastes (c_1, c_2) .

Un segment de forme est validé, si d'une part les points A, B, C, D existent, et si d'autre part les valeurs p_1, p_2 et c_1, c_2 sont supérieures respectivement à des seuils p_s et c_s . Les valeurs de ces seuils ont été choisies après de multiples essais, de façon à détecter correctement des formes très faiblement contrastées.

II.3.2. Prélocalisation par "voie matérielle" des sous-images

Cette prélocalisation a pour but essentiel de repérer les zones de l'image à fort contraste, incluant par conséquent les formes recherchées mais aussi les zones du contexte particulièrement contrastées (bords du support, par exemple).

Le principe de détection par "voie matérielle" consiste à rechercher, en temps réel, une image $T^*(i,j)$ représentant le mieux possible le contexte et d'en déduire les positions des zones contrastées par simple comparaison de cette image $T^*(i,j)$ et de l'image initiale $T(i,j)$

Une bonne réalisation de $T^*(i,j)$ est obtenue par filtrage bidimensionnel de type passe-bas, de l'image $T(i,j)$. En effet, les fréquences élevées filtrées représentent les zones de l'image à fort contraste.

Plus précisément, pour adapter cette méthode de détection au mode d'acquisition, c'est-à-dire ligne par ligne, le filtrage s'effectue pour j variable, à i constant. L'acquisition des pixels étant séquentielle, la variable j est directement liée à la variable temps : le filtrage de $T(i,j)$ se ramène simplement à un filtrage temporel. Dans notre application, $T^*(i,j)$ a été obtenue grâce à un filtre analogique du second ordre de type Butterworth.

La détection de contraste élevé est obtenue lorsque le signal représentant l'image $T^*(i,j) - T(i,j)$ dépasse un seuil prédéterminé T_s . La figure II.6 montre deux exemples de détection :

- pour une forme (cas 1)
- pour une zone de contexte à contraste élevé (cas 2), qui est éliminée ultérieurement par un traitement logiciel.

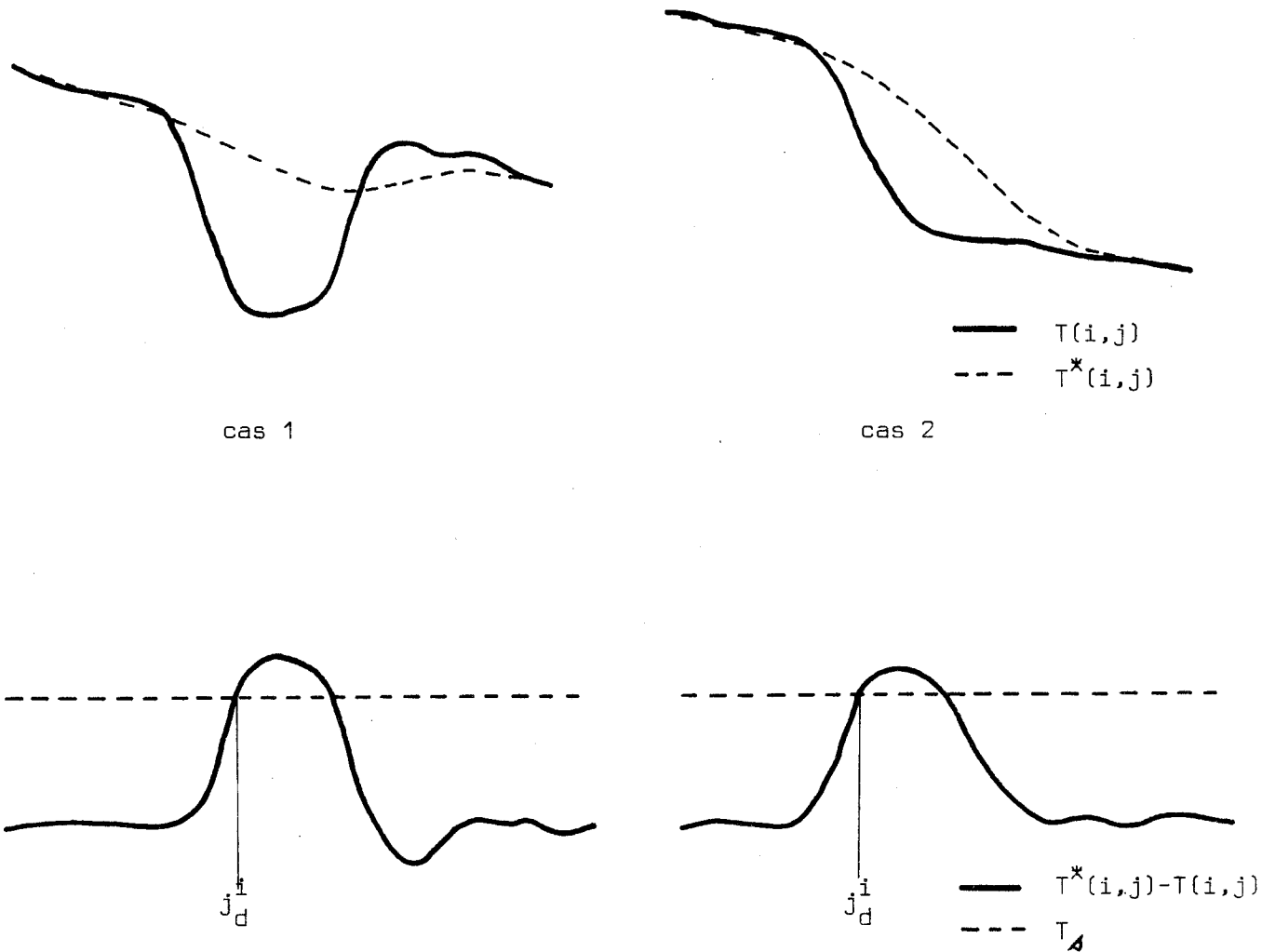


Figure II.6. : détection par
voie "matérielle"

II.3.3. Détection des formes par voie logicielle

II.3.3.1. Définition du segment de forme acquis

Avant de définir l'algorithme de recherche des points caractéristiques d'un segment de forme, il convient de définir le domaine d'investigation autour de la position j_d^i . Ce domaine doit être suffisant pour inclure en toute certitude les frontières gauche et droite d'un segment $s(i)$ quelconque. Pour cela, la recherche logicielle s'effectue dans un domaine $(j_d^i - \Delta_1, j_d^i + \Delta_2)$

- Δ_1 : étant donné que la détection par voie "matérielle" apparaît dans la zone frontière gauche, il est nécessaire d'acquérir les Δ_1 points d'image précédant $T(i, j_d^i)$ afin d'inclure totalement la frontière gauche.

- Δ_2 : cette valeur est choisie de telle sorte qu'une forme de largeur maximale (l_{max}) soit correctement acquise .

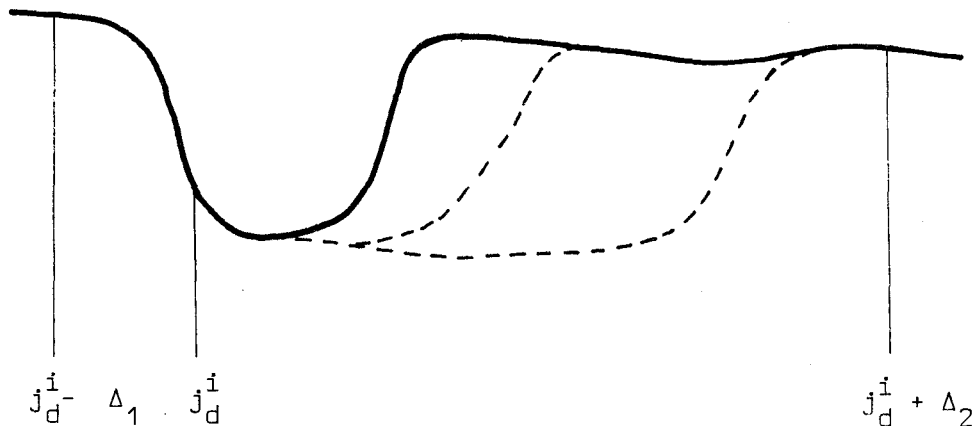


Figure II.7. : Domaine d'investigation
autour de j_d^i .

II.3.3.2. Paramétrisation des segments de forme

Pour obtenir une bonne détermination des points A, B, C, D dans le cas d'un signal d'image fortement bruité, il est préférable d'utiliser des méthodes itératives telles que la méthode des moindres carrés. Mais un tel procédé n'est pas envisageable dans notre application, car l'emploi de ces algorithmes suppose qu'il n'y ait aucune contrainte de vitesse d'exécution.

Une méthode très rapide, mais cependant sensible au bruit, consiste à analyser l'évolution des variations $\Delta T(i, j) = T(i, j) - T(i, j-1)$ en fonction de j . La présence d'un bruit moins important à l'intérieur de la forme que dans le contexte, nous a conduit à adopter l'algorithme suivant :

- détermination de l'intérieur de la forme

Cette localisation s'obtient par la recherche d'une position j_{int} telle que :

$$T(i, j_{int}) = \inf T(i, j); j \in (j_d^i - \Delta_1, j_d^i + \Delta_2) .$$

Cette définition permet de trouver une position $j \in (j_1, j_2)$

- détermination des frontières du segment de forme : j_1, j_2, j'_1, j'_2

Les zones frontières (j_1, j'_1) , (j_2, j'_2) se caractérisent par des valeurs $\Delta T(i, j)$ importantes. La détermination des points frontières est effectuée par comparaison des $\Delta T(i, j)$ à un seuil ΔTs .

* détermination de j'_1, j'_2

$$j'_1 = \sup j \text{ tel que } \Delta T(i, j) \leq \Delta Ts, \quad j \in (j_d^i - \Delta_1, j_{int})$$

$$j'_2 = \inf j \text{ tel que } \Delta T(i, j) \leq \Delta Ts, \quad j \in (j_{int}, j_d^i + \Delta_2)$$

* détermination de j_1, j_2

Pour éviter que la localisation de ces positions soit sensible au "bruit", les frontières gauche et droite sont déterminées lorsque les variations $\Delta T(i, j)$ restent faibles ($\leq \Delta Ts$) pour m valeurs adjacentes de j .

$$j_1 = \sup j \{ \Delta T(i, j-k) \leq \Delta Ts; \quad k \in (0, m); j-k \in (j_d^i - \Delta_1, j'_1) \}$$

$$j_2 = \inf j \{ \Delta T(i, j+k) \leq \Delta Ts; \quad k \in (0, m); j+k \in (j'_2, j_d^i + \Delta_2) \}$$

II.3.4. Exemples de recherche de segments de forme

a) cas peu bruité

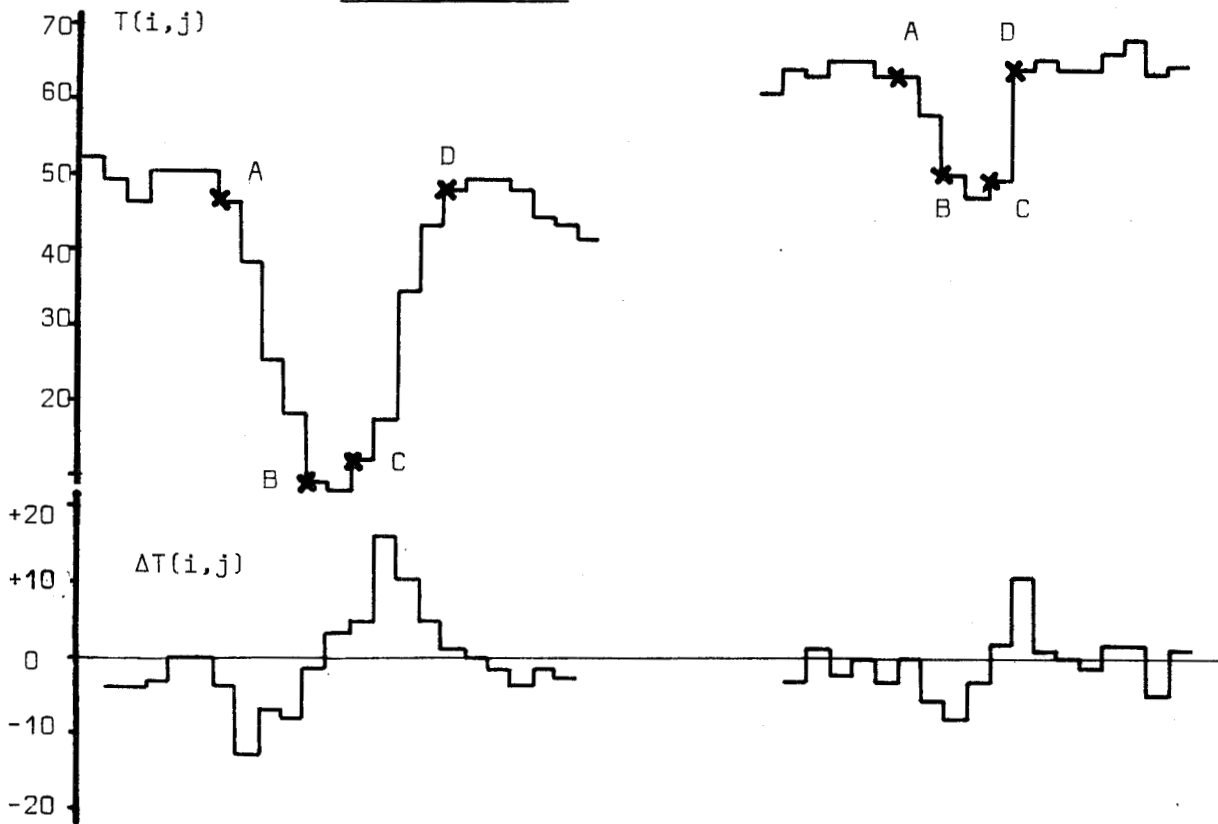


Figure II.8.a .

Figure II.8.b.

Les figures II.8a et II.8b montrent deux exemples de paramétrisation de segments de forme pour des cas peu bruités : on remarquera notamment la proximité entre les pentes calculées et les pentes réelles.

b) Paramétrisation de segments de forme très bruités

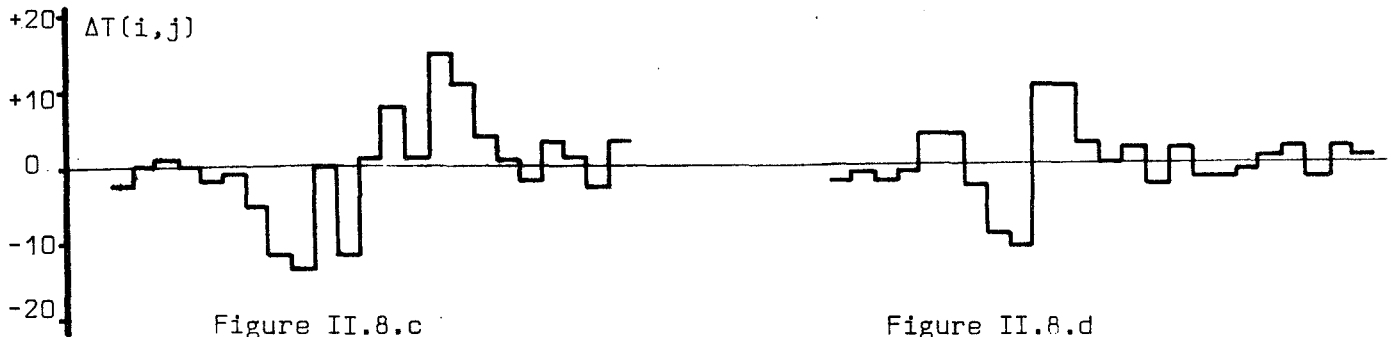
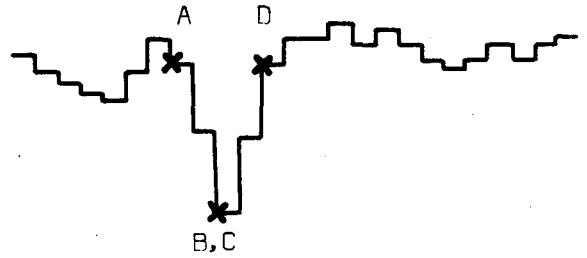
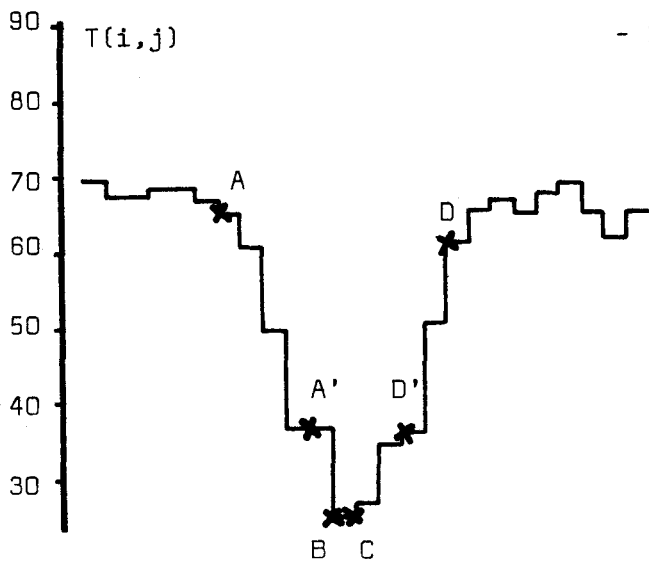


Figure II.8.c

Figure II.8.d

La figure II.8c justifie la nécessité de déceler les points A et D à partir de m tests sur les pentes élémentaires $\Delta T(i,j)$; un arrêt de la procédure au premier test négatif aurait conduit à placer ces points en A' et D' .

La figure II.8d représente le signal vidéo digitalisé correspondant à une forme de faible largeur. On constate que, malgré un contexte très bruité, la détermination de A, B, C et D est obtenue avec une précision satisfaisante.

c) cas de fausses détections

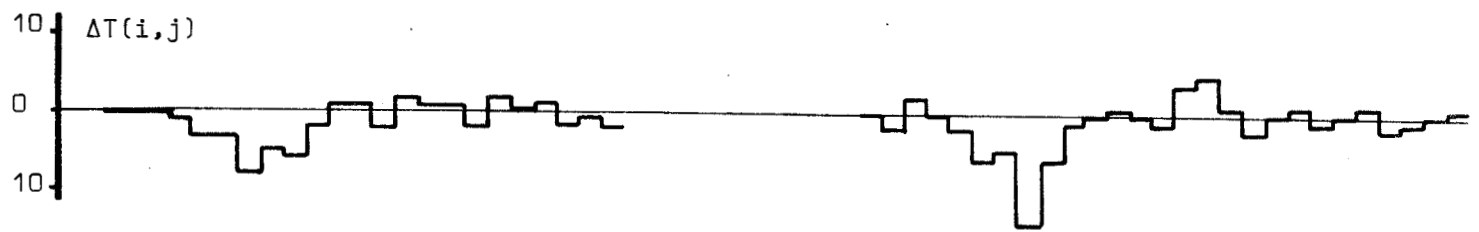
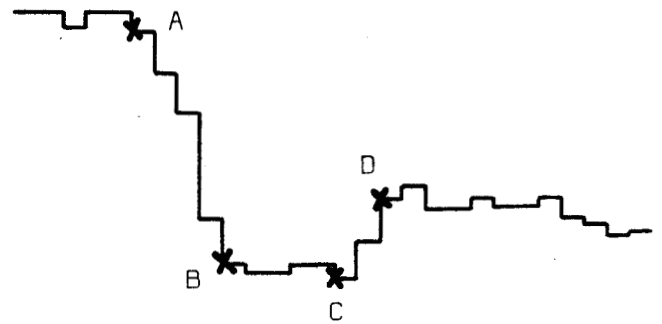
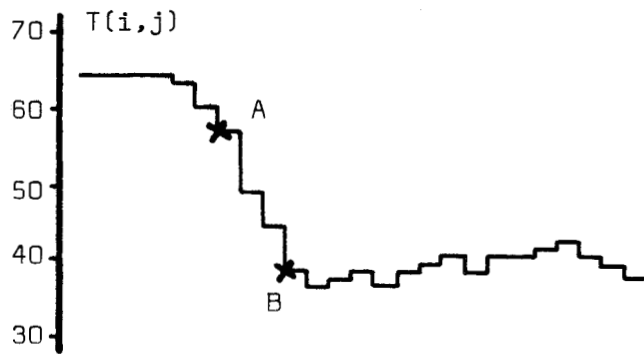


Figure II.8.e

Figure II.8.f

Ces deux cas proviennent de zones à contraste élevé apparues en bordure du contexte. La détection correspondant à la figure II.8e est éliminée car ce segment de forme ne peut être défini, par absence de frontière droite (inexistence de C).

Seules les pentes négatives (entre A,B) sont détectées grâce au procédé de pré-localisation par voie matérielle : cette remarque incite à valider l'existence de forme en testant, en priorité, la présence ou l'absence de frontière droite (points C et D).

Pour l'exemple de la figure II.8f, les points C et D ont été néanmoins décelés. Cette détection n'est pas validée car la pente p_2 est inférieure au seuil p_s .

II.4. Acquisition des sous-images par cadrage des formes

II.4.1. Relation de "connexité à priori"

La procédure décrite dans le sous-chapitre précédent permet de vérifier l'existence d'une forme par l'analyse d'un seul segment $s(i)$. Le segment traité est donc $s(i_1)$, i_1 étant la ligne où est apparue la première détection par voie "matérielle". Les segments des lignes suivantes $(i_1+1; i_2)$, i_2 étant la ligne de dernière détection, doivent vérifier la relation de connexité définie au §II.2.1, pour être intégrés à une même forme. Mais, avant de vérifier la connexité des segments, il faut que chaque segment soit ramené au triplet $\{i, j_1(i), j_2(i)\}$ $j_1(i), j_2(i)$ définissant les frontières gauche et droite (§II.2.1). Ce traitement, objet du chapitre III, nécessite un algorithme dont les temps d'exécution sont importants, et, par conséquent, ne peut être effectué au fur et à mesure de l'acquisition des segments. C'est pourquoi, il est nécessaire de définir une nouvelle relation de connexité qui ne fasse intervenir que les positions des détections par voie matérielle (j_D^i) .

Deux segments $s(i)$ et $s(i+1)$ sont "connectés à priori" si :

$$j_D^{i+1} \in [j_D^i - \Delta a, j_D^i + \Delta a]$$

Δa est appelé intervalle d'appartenance

II.4.2. Cadrage minimal des formes

Nous avons vu que sur une ligne d'indice i , l'acquisition du signal $T(i, j)$ pour des positions $j \in [j_D^i - \Delta 1, j_D^i + \Delta 2]$ permet d'inclure correctement les frontières gauche et droite d'une forme quelconque. Pour que le cadrage soit complet, il est cependant utile d'élargir le domaine acquis au delà des limites (i_1, i_2) , i_1 et i_2 étant respectivement les lignes sur lesquelles ont eu lieu la première et la dernière détection.

En effet, l'impulsion logique de détection issue de la partie "matérielle" n'apparaît que lorsque la pente du signal vidéo est suffisante.

On risque donc de détecter une forme sur une ligne, i_1 , alors que celle-ci apparaît déjà faiblement sur les lignes $[i_1 - n_a, i_1 - 1]$. Il est donc nécessaire, pour cadrer correctement les formes, de connaître les données $T(i-k, j)$, $j \in [j_d^i - \Delta_1, j_d^i + \Delta_2]$, $k \in [1, n_a]$. Ceci suppose donc que le système d'acquisition soit capable de mémoriser les lignes $i - n_a$ à $i - 1$ lorsque la ligne i est balayée. La valeur choisie, $n_a = 2$, permet un cadrage correct, pour la totalité des cas analysés.

La procédure d'acquisition peut alors se décomposer comme suit :

- initialisation de l'acquisition d'une sous-image déclenchée par la validation logicielle d'une détection au point de l'image $(i_1, j_d^{i_1})$. Les segments acquis sont composés des données :

$$T(i_1 - k, j), j \in [j_d^{i_1} - \Delta_1, j_d^{i_1} + \Delta_2]$$

$$k \in [0, 2]$$

- poursuite de l'acquisition sur une ligne $i_1 + k$

Le segment de forme, composé des données $T(i_1 + k, j)$, $j \in [j_d^{i_1 + k} - \Delta_1, j_d^{i_1 + k} + \Delta_2]$, est inclus à la sous-image si la relation de "connexité à priori" est vérifiée :

$$j_d^{i_1 + k} \in [j_d^{i_1 + k - 1} - \Delta_a, j_d^{i_1 + k - 1} + \Delta_a]$$

- fin de l'acquisition à la ligne i_2

La fin de la forme est déterminée lorsqu'**aucune** détection n'est apparue sur les lignes $i_2 + 1, i_2 + 2$ dans l'intervalle $[j_d^{i_2} - \Delta_a, j_d^{i_2} + \Delta_a]$. La remarque concernant le cadrage du haut de la forme est valable aussi pour la fin : la dernière détection a eu lieu sur la ligne d'indice i_2 alors que la forme peut ne pas être totalement terminée. Par conséquent, le cadrage complet est accompli après acquisition des segments composés des données $T(i_2 + k, j)$, $j \in [j_d^{i_2} - \Delta_1, j_d^{i_2} + \Delta_2]$, $k \in [0, 2]$.

II.4.3 Exemples d'acquisition de sous-images

Les figures II.9 et II.10 montrent le résultat de l'acquisition de sous-images contenant respectivement une forme compacte et une forme allongée . Pour des raisons liées à la présentation des résultats, les figures II.9a et II.10a ne représentent pas totalement les sous-images acquises. En effet, chaque segment de forme est tronqué : $j \in [j_D^i - \delta_1, j_D^i + \delta_2]$; $\delta_1 < \Delta_1$; $\delta_2 < \Delta_2$.

Les visualisations des sous-images acquises (figures II.9b et II.10b) permettent un contrôle visuel du cadrage correct des formes. Nous avons adapté une échelle de niveaux de gris composée de caractères alphanumériques pour chaque sous-image. En effet, les transparences d'une forme et du contexte sont variables d'un exemple à l'autre. Cette adaptation s'effectue en tenant compte des valeurs extrêmes de $T(i,j)$: le caractère alphanumérique le plus foncé correspond aux valeurs élevées de $T(i,j)$, un caractère clair correspond aux valeurs faibles.

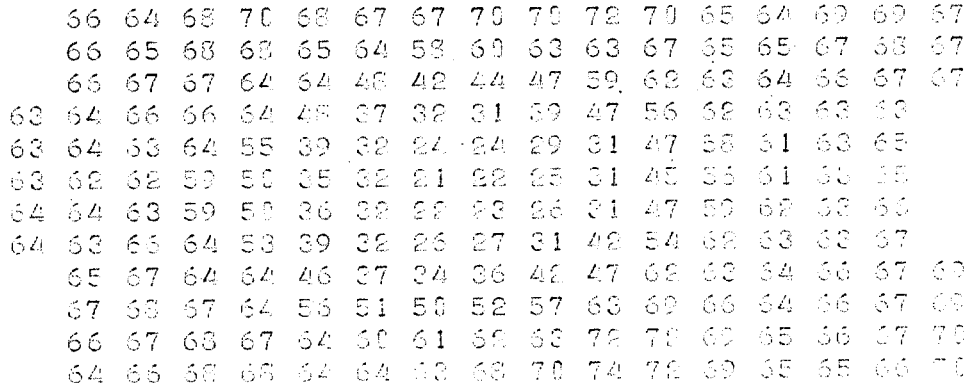


Figure II.9a

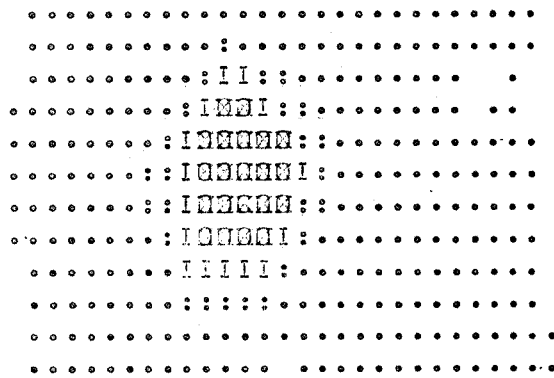


Figure II.9b

Figure II.9 : Sous-image d'une forme compacte

```

76 75 73 75 71 68 67 72 70 70 72 73 73 74 74 75
79 77 72 73 69 64 57 63 63 69 72 71 72 72 75 77
79 73 72 71 64 48 27 47 60 63 71 71 72 73 77 78
79 79 78 72 70 64 36 39 43 58 63 70 70 74 76 77
79 77 71 71 64 37 32 43 57 63 71 71 75 77 77 76
79 74 70 72 64 37 32 41 56 63 70 71 74 77 77 76
76 76 72 72 72 64 36 32 31 56 63 68 71 75 77 76
74 73 73 74 64 38 32 31 47 63 67 70 75 75 76 73
74 73 74 74 64 39 32 37 47 63 68 69 73 73 76 77
72 74 73 64 39 32 37 47 53 68 70 72 73 76 73 76
74 72 74 73 64 39 32 37 47 63 69 72 74 74 76 75
74 72 73 75 64 40 32 35 47 63 68 73 74 74 76 79
75 73 74 75 64 40 32 36 47 63 68 74 75 74 76 77
75 74 74 74 64 40 32 31 47 63 63 73 76 74 75 75
74 73 73 75 64 41 32 31 47 61 63 73 76 74 77 76
73 72 73 75 64 41 32 31 47 61 63 74 75 74 77 74
73 71 73 74 64 42 32 32 45 60 63 74 73 76 77 72
74 72 73 73 64 41 32 32 44 59 63 72 71 77 75 72
75 71 73 74 64 43 32 32 31 56 63 71 70 75 76 72
76 71 73 72 64 43 32 32 31 55 63 70 70 74 76 72
77 70 71 71 64 44 33 32 40 47 63 70 70 75 76 72
73 71 72 72 66 64 50 47 47 59 33 71 70 73 76 72
73 72 73 74 70 64 66 69 69 67 74 72 71 73 75 73
78 73 73 74 69 65 69 73 75 72 74 73 71 74 76 74

```

Figure II.10a

```

. . . . .
. . . . . : : : . . . . .
. . . . . : I I I : . . . . .
. . . . . : H H I : . . . . .
. . . . . : O O I : . . . . .
. . . . . : S S S : . . . . .
. . . . . : A A A : . . . . .
. . . . . : Q Q Q I : . . . . .
. . . . . : T T T I : . . . . .
. . . . . : M M I : . . . . .
. . . . . : S S S I : . . . . .
. . . . . : S S S I : . . . . .
. . . . . : S S S I : . . . . .
. . . . . : S S S I : . . . . .
. . . . . : S S S I : . . . . .
. . . . . : S S S I : . . . . .
. . . . . : S S S I : . . . . .
. . . . . : I S S S : . . . . .
. . . . . : I S S S : . . . . .
. . . . . : I S S S I : . . . . .
. . . . . : : I I I : . . . . .
. . . . . : : : : . . . . .
. . . . . : : : : . . . . .

```

Figure II.10b

Figure II.10 :
Sous-image d'une
forme allongée.

II.5. Conclusion

La procédure d'acquisition des formes en temps-réel ne constitue qu'une première étape d'un processus qui doit aboutir à la reconnaissance des formes. Le résultat de cette acquisition est un ensemble de sous-images extraites de l'image complète, sans aucune perte d'information sur les formes. La représentation imagée des formes n'est pas adaptée à un traitement de reconnaissance. En effet, les sous-images, comme toute image en général, sont redondantes d'informations.

La phase qui va suivre, objet du chapitre III, a pour but de ramener ces informations à un volume plus petit, sans perte d'informations utiles à la reconnaissance. Cette réduction consiste à ne retenir des sous-images que les contours ou lignes séparatrices des formes et de leur contexte.

C H A P I T R E III

- RECHERCHE DE CONTOUR DES FORMES

Le traitement des sous-images, visant à extraire le contour des formes, doit posséder certaines propriétés pour répondre aux impératifs définis au chapitre II.

- La procédure doit être rapide : nous développerons une méthode de traitement séquentiel des sous-images, c'est-à-dire ligne par ligne, permettant ainsi une acquisition et une détection de contour quasi simultanées.

- La méthode doit, d'une part, être très peu sensible au "bruit d'image"; d'autre part, la recherche de contour ne doit pas être perturbée par d'éventuels changements de transparence soit du contexte soit de la forme elle-même.

La connexité des segments de forme est redéfinie dans ce chapitre, affinant donc la relation de "connexité à priori" établie dans le chapitre précédent.

III.1. Généralités sur la recherche de contour

III.1.1. Techniques de recherche de contour

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

La littérature scientifique est très riche en techniques de recherche de contour; toutefois , il est possible de dégager quelques principes généraux :

- Méthode du gradient : (35,36,37,38)

Les zones frontières des formes se différencient des autres régions de l'image par des variations importantes d'intensité lumineuse. Pour formaliser cette notion, le concept de gradient vectoriel a été introduit.

Pour une image $T(x,y)$ supposée continue en x et y , on définit le vecteur gradient par :

$$G = \begin{cases} \frac{\partial T(x,y)}{\partial x} \\ \frac{\partial T(x,y)}{\partial y} \end{cases} \quad \text{ou } G = g e^{-i\theta} \text{ (forme polaire)}$$

La ligne de contour ou ligne de séparation entre forme et contexte peut être définie comme la ligne des crêtes du module g du gradient. Pour chaque point de la ligne de contour appelé point frontière, la direction définie par l'angle θ est perpendiculaire à la tangente au contour,

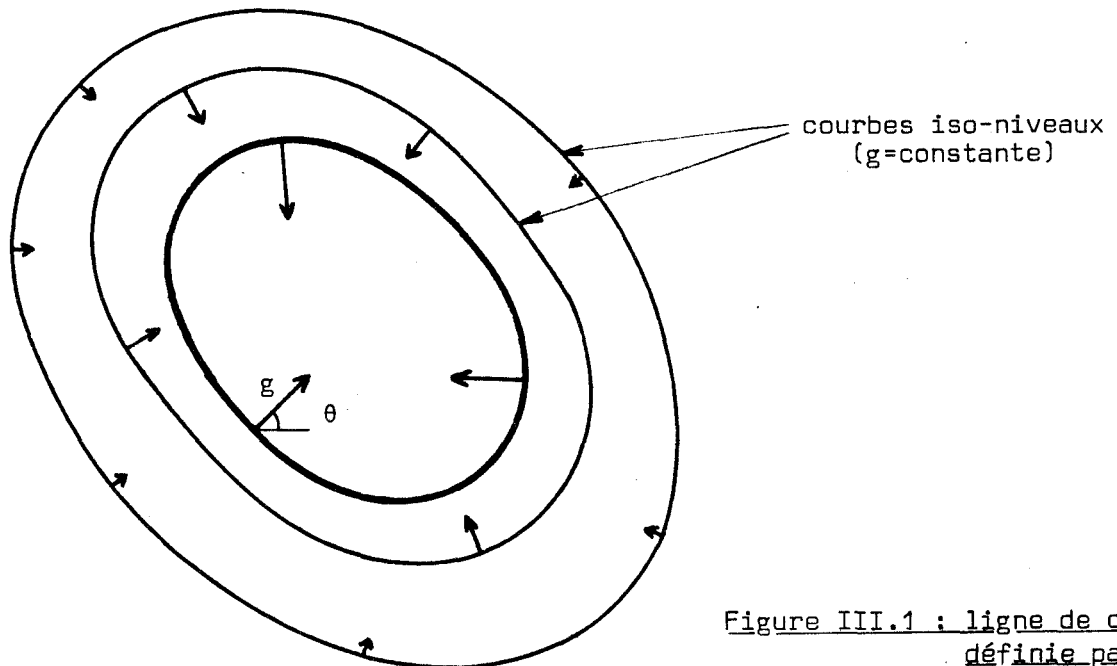


Figure III.1 : ligne de contour définie par le gradient.

Les variantes des méthodes utilisant le gradient, apparaissent lors du passage d'une image $T(x,y)$ continue en x et y , à une image digitalisée $T(i,j)$. Lorsque l'image est fortement bruitée, il est souhaitable que le gradient en un point soit calculé à partir d'une "fenêtre" encadrant le point considéré (opérateurs de SOBEL, HUECKEL, ROBERTS.).

- Méthode à base statistique (39,40,41)

Les méthodes statistiques de détection de contour se basent sur l'analyse de la répartition des fréquences d'apparition des intensités lumineuses, obtenue par un histogramme (Figure II.2.) . La forme dont on recherche le contour se caractérise par des niveaux de gris différents de ceux du contexte, l'histogramme fait alors apparaître deux modes ou "pics" caractérisant la forme et le contexte. La vallée située entre les "pics" permet donc de séparer la forme de son contexte. Ainsi, le contour est obtenu en seuillant l'image par la valeur T_v du fond de la "vallée".

En utilisant cette méthode, les difficultés apparaissent lors de la détermination de la "vallée" par l'analyse de l'histogramme : lorsque les images sont bruitées, la présence de minima locaux rend difficile la localisation de T_v .

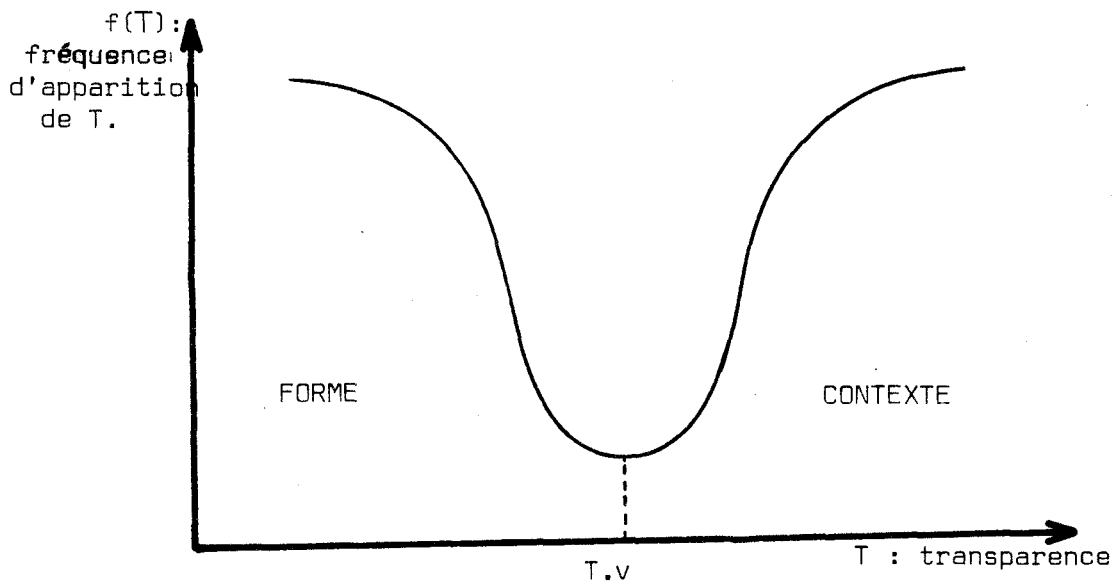


Figure III.2. : Histogramme de densité de niveaux de gris

III.1.2. Utilisation de ces méthodes

La détermination des points frontières par l'exploitation du gradient calculé en chaque point des sous-images, est dans notre cas, un moyen inadapté aux impératifs de temps réel. En effet, la présence de bruit dans l'image, nécessite un calcul de gradient à partir d'une "fenêtre" assez large autour de chaque point considéré, et ainsi les temps de calcul deviennent prohibitifs.

De même, une méthode reposant sur l'analyse des histogrammes de densité de niveaux de gris, n'est pas exploitable. Certaines formes, de même que les contextes associés, n'ont pas une transparence constante dans toute la sous-image. La détermination des frontières par un seuil unique T_v est, dans ces conditions, inadaptée au problème. Pour éviter cet inconvénient il est possible de traiter les sous-images par parties (quelques lignes par exemple) . Mais la détermination de T_v pour chacune de ces parties serait entachée d'erreurs, car l'analyse porterait sur un nombre trop faible de pixels. Cette remarque est également valable lors de la recherche de contour de formes de petite taille.

III.2. Recherche de contour par seuils adaptatifs

Le principe de la méthode est d'analyser les sous-images ligne par ligne . Pour un segment de forme $s(i)$, la recherche de contour se ramène donc à la détermination des points frontières gauche (C_g^i) et droit (C_d^i).

Ce sous-chapitre est consacré à l'exposé de la méthode , qui à partir d'une définition simple dérivée du concept de gradient vectoriel, aboutit à la notion de seuils calculés pour chaque segment de forme.

III.2.1. Principe de la détection de contour

Lorsque les sous-images sont analysées ligne par ligne, le gradient en un point se ramène à la dérivée par rapport à j , i étant constant. Les positions C_g^i et C_d^i sont donc détectées en recherchant les extrema des variations $\Delta T(i,j) = T(i,j) - T(i,j-1)$:

C_g^i est tel que $\Delta T(i, C_g^i) \leq \Delta T(i, j)$

C_d^i est tel que $\Delta T(i, C_d^i) \geq \Delta T(i, j)$

$$j \in [j_D^i - \Delta 1, j_D^i + \Delta 2]$$

La présence de bruit dans les sous-images se répercutant sur les pentes élémentaires $\Delta T(i, j)$, rend inexploitable une telle définition.

Lorsque le segment $s(i)$ est représenté, non plus par les pixels $T(i, j)$, mais par les positions caractéristiques j_1, j_1', j_2, j_2' (§II2.3) il est plus aisé de définir les points frontières, car les secteurs j_1, j_1' et j_2, j_2' délimitent les zones frontières de la forme. Rappelons qu'une telle représentation est obtenue par application des définitions du §II3.3.2.

Considérons un segment de forme peu bruité, dont les positions caractéristiques sont déterminées avec précision (figure III.4). Dans ce cas les points B' et C', sont confondus, ou tout au moins très proches des points à dérivée maximale.

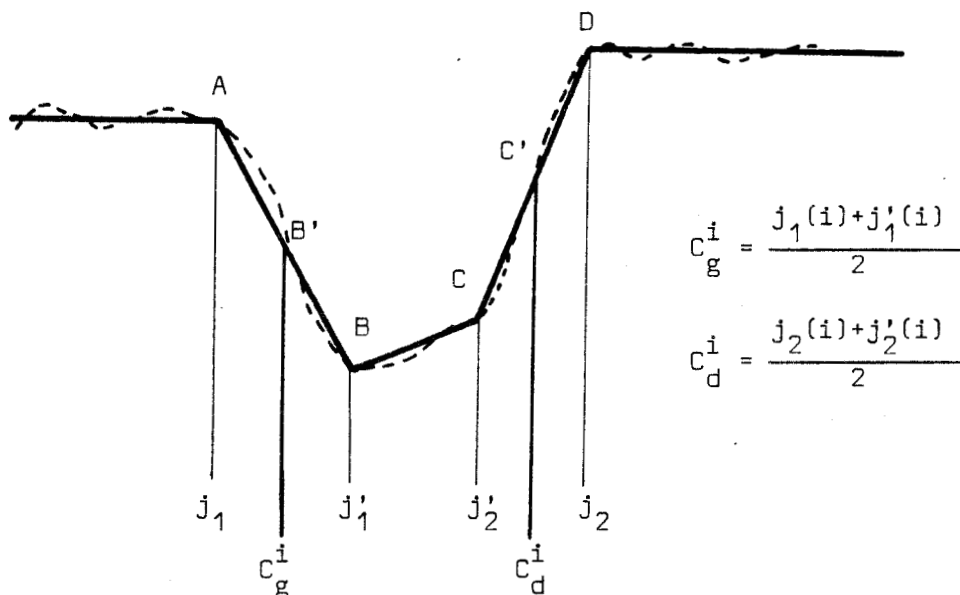


Figure III.4 : Représentation des points frontières comme milieux des zones frontières.

Toutefois, l'utilisation de ces définitions a montré que la détermination des frontières, est assez sensible au bruit d'image. Cette constatation est notamment mise en évidence lors de l'extraction de contour d'une forme aux bords parfaitement rectilignes. Le contour trouvé est alors une ligne brisée, faisant apparaître, ce que nous appellerons, un bruit de contour.

III.2.2. Recherche des frontières par seuillage adaptatif.

Une simple amélioration permet de réduire considérablement le bruit de contour. Au lieu de définir les frontières gauche et droite par les milieux des segments $[j_1, j'_1]$ et $[j_2, j'_2]$ introduisons la notion de seuils gauche S_g^i et droit S_d^i . Ces valeurs sont calculées à partir des transparences aux points A, B, C et D.

$$S_g^i = \frac{T(i, j_1) + T(i, j'_1)}{2}$$

$$S_d^i = \frac{T(i, j_2) + T(i, j'_2)}{2}$$

Les frontières s'obtiennent en seuillant $T(i, j)$ en partant de la position j_{\min}^i :

$$j_{\min}^i \text{ est tel que } T(i, j_{\min}^i) = \inf T(i, j)$$

$$j \in [j_d^i - \Delta 1, j_d^i + \Delta 2]$$

$$C_g^i = \sup j \text{ tel que } T(i, j) \geq S_g^i$$

$$j \in [j_d^i - \Delta 1, j_{\min}^i]$$

$$C_d^i = \inf j \text{ tel que } T(i, j) \geq S_d^i$$

$$j \in [j_{\min}^i, j_d^i + \Delta 2]$$

Remarquons que cette définition est similaire à la définition précédente (§III.2.1), lorsque le segment de forme est peu bruité.

La figure III.5 montre un exemple de segment de forme bruité où les positions des points A,B,C et D sont imprécises. Néanmoins les transparences $T(i,j)$ et $T(i,j')$ sont très proches des transparences respectivement du contexte et de la forme. Par conséquent la valeur du seuil

$S_g = \frac{T(i,j_1) + T(i,j'_1)}{2}$ est très peu sensible au "bruit".

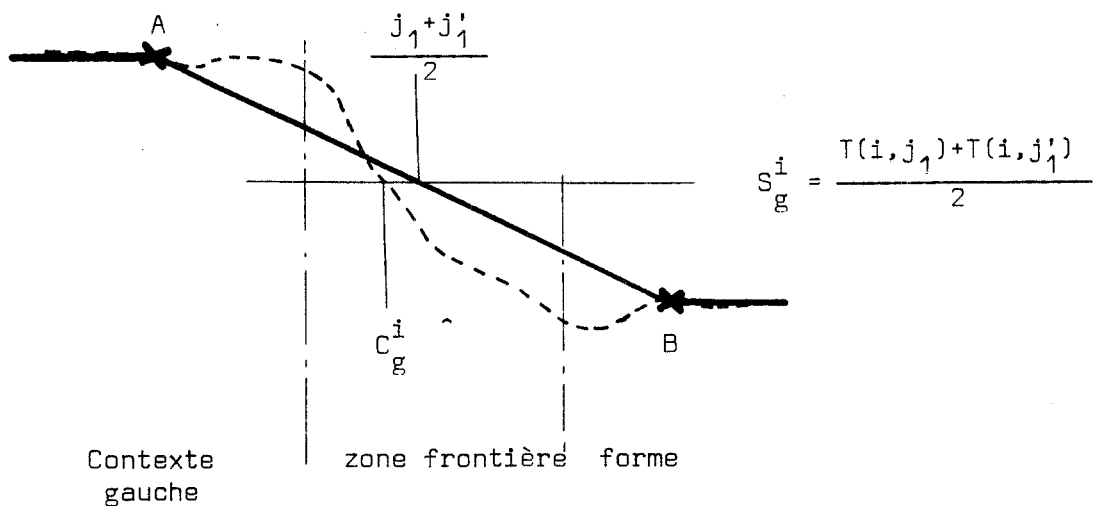


Figure III.5. : Détermination d'une frontière gauche grâce au seuil S_g^i .

Cette amélioration de la définition des points frontières n'élimine que partiellement le bruit de contour. Ce bruit est mis en évidence en étudiant l'évolution des seuils S_g^i et S_d^i des segments $s(i)$ d'une forme, se présentant dans un contexte uni, et dont les intensités lumineuses à l'intérieur de la forme sont constantes quelque soit le segment étudié. Dans ces conditions, les seuils droits et gauches devraient être également constants pour toute la sous-image. Les résultats ont montré des variations sensibles de S_g^i et S_d^i , conséquence d'une détermination rapide, donc imprécise des points A, B, C et D.

III.2.3. Lissage des seuils adaptatifs

Pour réduire les variations trop brutales des seuils S_g^i et S_d^i et, par conséquent, le bruit de contour, nous effectuons un lissage, obtenu par filtrage numérique du type passe-bas (1er ordre).

$$S_{g \text{ ou } d}^{*i} = a S_{g \text{ ou } d}^i + b S_{g \text{ ou } d}^{*i-1}; a+b = 1$$

- $S_{g \text{ ou } d}^i$: valeur du seuil calculée pour un segment $s(i)$
(§III.2.1.)

- $S_{g \text{ ou } d}^{*i}$: valeur filtrée, utilisée pour la détermination des points frontières.

- Le choix de "a" et "b" résulte d'un compromis entre un filtrage important ($a < b$), conduisant à une réduction suffisante des variations brutales des seuils, et un filtrage faible ($a > b$) permettant de lentes variations des seuils lors d'un changement de transparence de la forme ou du contexte.

L'expérience a montré que le choix $a=b=0,5$ est un bon compromis. D'autre part, la simplicité de programmation de ce filtre et, par conséquent, la rapidité d'exécution facilitent la réalisation en temps réel de la recherche de contour.

III.3. Adaptation aux sous-images acquises

La méthode de recherche de contour par seuils adaptatifs est définie pour un segment $s(i)$ d'une forme $i \in [i_1, i_2]$, où i_1 et i_2 sont respectivement les indices de ligne de la première et dernière détection par voie "matérielle". Cette méthode ne peut s'appliquer sur les segments du début de la forme $s(i_1-1)$, $s(i_1-2)$ et de fin de forme $s(i_2+1)$, $s(i_2+2)$.

En effet, l'absence de contrastes élevés sur ces segments conduit à une mauvaise détermination des points A,B,C et D et entraîne, par conséquent, une erreur sur les contours de "début" et de "fin" de forme.

L'investigation des sous-images se déroule comme suit :

a) Traitement des segments $s(i_1)$, $s(i_1-1)$, $s(i_1-2)$.

Les seuils S_g^i et S_d^i calculés à partir du segment $s(i_1)$ permettent de rechercher les frontières $s(i_1-2)$, $s(i_1-1)$, $s(i_1)$ en effectuant un seuillage des données $T(i,j)$.

Certaines précautions doivent cependant être prises, car le contexte environnant la forme peut présenter des variations telles que certaines zones extérieures à la forme possèdent des valeurs $T(i,j)$ inférieures aux seuils. La figure III.6. montre un exemple de ce type, où la frontière $C_d^{i_1-2}$ est mal positionnée.

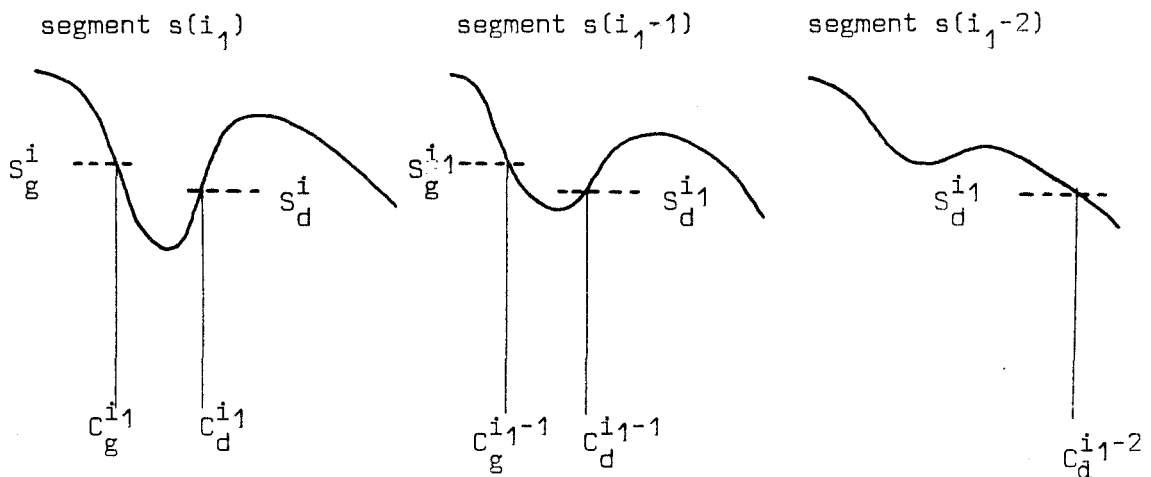


Figure II.6. : Exemple d'erreur de détermination de $C_d^{i_1-2}$

Pour éliminer ce risque d'erreur, la zone d'investigation ou de recherche des points frontières est limitée à une fenêtre j_{gf}, j_{df} définie par les frontières du segment $s(i_1)$.

$$j_{gf} = C_g^{i_1} - \Delta f ; j_{df} = C_d^{i_1} + \Delta f.$$

f est une valeur suffisamment petite pour ne pas perdre d'informations sur la forme.

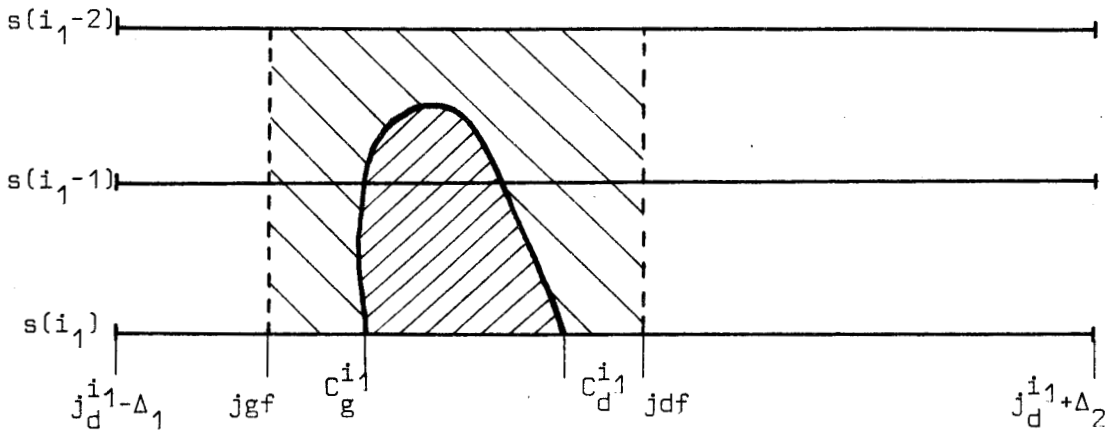


Figure III.7. : Définition de la fenêtre d'investigation

La procédure totale se résume par les étapes suivantes :

- calcul des seuils $S_g^{i_1}$, $S_d^{i_1}$

$$S_g^{i_1} = \frac{T(i_1, j_1^{i_1}) + T(i_1, j_1^{i_1})}{2} ; \quad S_d^{i_1} = \frac{T(i_1, j_2^{i_1}) + T(i_1, j_2^{i_1})}{2}$$

- recherche des points frontières du segment $s(i_1)$

$$c_g^{i_1} = \sup_{j \in [j_d^{i_1} - \Delta_1, j_{\min}^{i_1}]} j \text{ tel que } T(i_1, j) \geq S_g^{i_1}$$

$$c_d^{i_1} = \inf_{j \in [j_{\min}^{i_1}, j_d^{i_1} + \Delta_2]} j \text{ tel que } T(i_1, j) \geq S_d^{i_1}$$

- recherche des points frontières des segments $s(i_1-1)$, $s(i_1-2)$

$$\left. \begin{aligned} C_g^{i_1-k} &= \sup j \text{ tel que } T(i_1-k, j) \geq S_g^{i_1} \\ &\quad j \in [C_g^{i_1} - \Delta f, j_{\min}^{i_1}] \\ C_d^{i_1-k} &= \inf j \text{ tel que } T(i_1-k, j) \geq S_d^{i_1} \\ &\quad j \in [j_{\min}^{i_1}, C_d^{i_1} + \Delta f] \end{aligned} \right\} k = 1 \text{ et } 2$$

b) traitement d'un segment $s(i)$, $\epsilon]i_1, i_2]$: les seuils S_g^i et S_d^i sont calculés après détermination des positions caractéristiques : $j_1^i, j_1'^i, j_2^i, j_2'^i$.

$$S_g^i = \frac{1}{2} \left[\frac{T(i, j_1^i) + T(i, j_1'^i)}{2} + S_g^{i-1} \right]$$

$$S_d^i = \frac{1}{2} \left[\frac{T(i, j_2^i) + T(i, j_2'^i)}{2} + S_d^{i-1} \right]$$

$$C_g^i = \sup j \text{ tel que } T(i, j) \geq S_g^i, \quad j \in [j_d^i - \Delta 1, j_{\min}^i]$$

$$C_d^i = \inf j \text{ tel que } T(i, j) \geq S_d^i, \quad j \in [j_{\min}^i, j_d^i + \Delta 2]$$

c) traitement des segments de "fin" de sous-images $s(i_2+1)$, $s(i_2+2)$

La procédure utilisée est identique à la procédure employée pour les segments de "début" de la sous-image : détermination d'une fenêtre à partir des frontières $C_g^{i_2}$, $C_d^{i_2}$, recherche de contour, dans la fenêtre, pour les segments $s(i_2+1)$, $s(i_2+2)$

III.4. Connexité des segments

III.4.1. Insuffisance de la connexité "à priori"

Nous avons défini au chapitre II (§4.1) une relation de connexité "à priori" des segments, établie uniquement à partir des positions j_d^i des détections par voie "matérielle". Cette relation permet d'associer les segments d'une même forme lors de l'acquisition des sous-images. Une telle définition de la connexité devient insuffisante lorsqu'une forme est suivie d'une zone du contexte à contraste élevé provoquant des détections "matérielles" qui, vérifiant la relation de connexité, sont intégrées à la forme. La figure III.8 montre un exemple d'un tel cas particulier et les conséquences sur la détermination du contour par la procédure exposée précédemment.

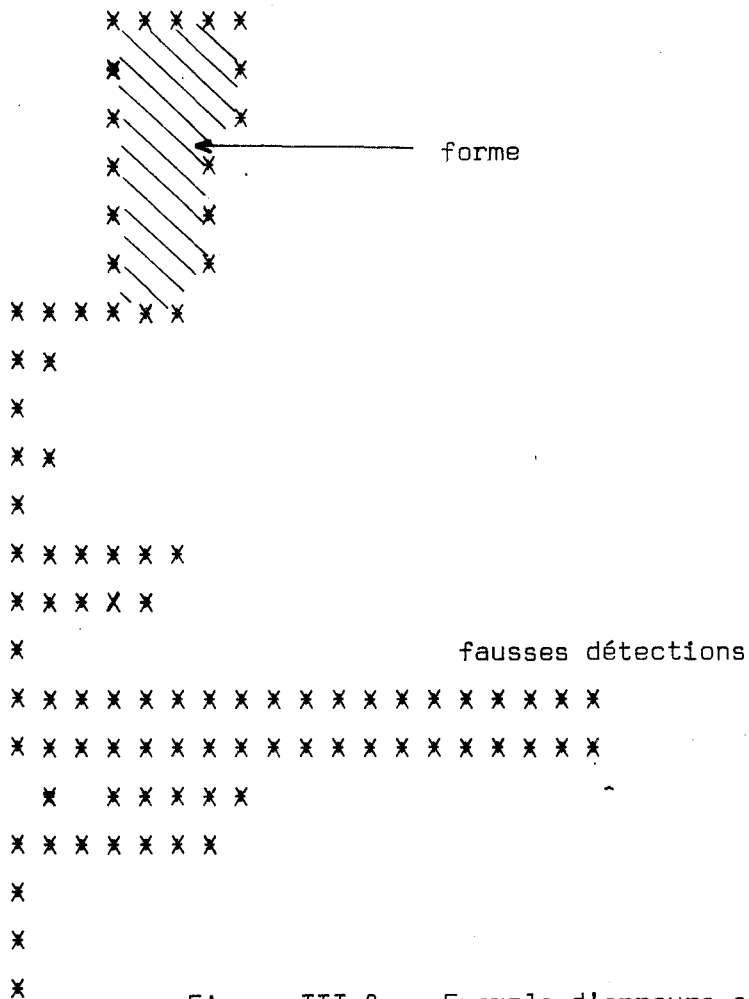


Figure III,8. : Exemple d'erreurs sur le contour d'une forme.

Evidemment, il n'est pas possible de réagir au niveau de l'acquisition des sous-images à cause des impératifs de temps réel. Ce cas est donc étudié lors de la recherche de contour.

III.4.2. Amélioration de la définition de connexité des segments

Afin de repérer d'éventuels segments $s(i)$ qui ne sont dûs qu'à la présence d'une zone très contrastée du contexte, chaque segment d'une sous-image est considéré, à priori, comme une présomption de segment de forme. Les tests définis dans le chapitre II (§II.2.3) sont donc appliqués à chaque segment $s(i)$. Si les pentes p_1 , p_2 et contrastes C_1 , C_2 sont supérieurs respectivement aux seuils p_s et c_s , la recherche de contour est effectuée en suivant la procédure normale (§III.3b). Dans le cas contraire, on considère que le segment $s(i)$ est un segment de fin de défaut.

Par conséquent, la recherche des frontières s'effectue comme pour les segments $s(i_2+1)$ $s(i_2+2)$ (§III.3.c). On est alors placé devant l'alternative suivante :

a) si les frontières sur le segment $s(i)$ existent, la forme est considérée comme non terminée, et les segments d'indice supérieur $s(i+1)$ $s(i+2)$ subissent le même traitement que le segment $s(i)$.

C'est le cas notamment d'une forme comportant une partie de faible largeur. Les contrastes et pentes sont alors très faibles.

b) si les frontières n'existent pas, la forme est donc complètement terminée. Les segments suivants $s(i+1)$, $s(i+2)$... sont alors considérés comme des présomptions de forme et les tests de validation des détections "matérielles" sont appliqués pour chacun de ces segments. Si le test est négatif, le segment est éliminé; si au contraire, la détection est validée, nous considérons qu'une nouvelle forme est apparue et la procédure de recherche de contour est à nouveau appliquée.

Cette amélioration de la définition de la connexité des segments a permis non seulement l'élimination de fausses détections qui au préalable étaient incluses à une forme, mais aussi l'élimination d'erreurs en fin de certaines formes. En effet, dans certains cas, le segment $s(i_2)$ est très faiblement contrasté, entraînant des erreurs sur le calcul des seuils et par conséquent sur le contour de la forme.

III.5. Résultats de la recherche de contour

Les exemples de recherche de contour sur quelques formes ont été choisis pour mettre en évidence l'efficacité des méthodes employées ainsi que l'opportunité de certaines modifications du principe général.

III.5.1. Détection de contour d'une forme compacte et d'une forme allongée.

Les contours des figures III.9.a et III.9.b sont le résultat du traitement d'image effectué respectivement pour une forme allongée et une forme compacte . Les sous-images de ces formes ainsi que leur représentation par des caractères alphanumériques figurent dans le chapitre précédent (figure II.9 et II.10). Les valeurs des seuils S_g^{*i} et S_d^{*i} sont également portées sur la figure III.9.

Cette présentation des résultats intermédiaires met en évidence la régularité des seuils de la forme allongée (figure III.9.a). Ce résultat était attendu car cette forme et son contexte ont des transparences quasi constantes pour toute la sous-image.

D'autre part, la régularité du bord gauche que l'on peut déceler après examen de la figure II.10.a est totalement retrouvée sur le contour de la figure III.9.a,

L'exemple de recherche de contour sur une forme compacte (figure III.9.b) justifie la modification développée dans la partie III.4. Sur l'exemple traité, la ligne i_2 qui est la ligne de dernière détection par voie "matérielle" présente des contrastes trop faibles ($c_1, c_2 < c_s$). La recherche de contour s'est donc prématurément terminée sur la ligne i_2-1 .

III.5.2. Histogrammes

Pour chacune des deux formes citées précédemment, les histogrammes de densité de niveaux de gris ont été tracés (figure III.10.a : forme compacte , figure III.10.b : forme allongée).

La valeur moyenne des seuils calculés sur toute la sous-image est également portée sur chacun de ces tracés . Nous ne retrouvons pas directement sur ces histogrammes la "vallée" située entre deux "pics". En ce qui concerne la forme compacte, seul le "pic" caractéristique du contexte de la sous-image est apparent pour des valeurs de transparences supérieures à 60 (figure III.10.a). Par contre, l'histogramme tracé à partir de l'image de la forme allongée (figure III.10.b) fait apparaître, à défaut de deux "pics", deux régions séparées par une zone où la densité de transparence est nulle.

Cette zone constitue la "vallée" . Il est à remarquer que la valeur moyenne des seuils droits et gauches se situe dans cette "vallée".

III.5.3. Seuils adaptatifs

Les figures III.11, 12, 13 et 14 ont pour but de montrer la nécessité de constituer des seuils adaptatifs et de les filtrer (§III.2.2). La forme choisie présente des variations importantes de transparence. Ceci se remarque sur la figure III.12 grâce au dessin de la forme, constitué de caractères alphanumériques. Cette figure comporte également les seuils S_g^{*i} et S_d^{*i} ainsi que les pentes p_1, p_2 pour chacun des segments $s(i)$. Pour des raisons de programmation liées à la précision des calculs, les résultats inscrits sont en fait $8xp_1, 8xp_2$.

Le contour de la forme a été représenté d'une part sur la figure III.13, mais aussi directement dans la sous-image de la figure III.11, ainsi que sur le dessin de la figure III.12.

Les évolutions des seuils en fonction de l'indice de ligne sont représentées sur la figure III.14.

L'image de la forme est partagée en trois zones A, B et C, permettant ainsi une meilleure analyse des résultats.

ZONE A : les seuils gauches S_g^i de cette zone sont différents des seuils droits. En effet, la transparence du contexte n'est pas homogène. La transparence moyenne du contexte gauche est égale à 68, celle du contexte droit est égale à 78.

ZONE B : cette zone se différencie de la zone A par une augmentation de la transparence de la forme. Pour tenir compte de ces variations les seuils ont des valeurs inférieures aux seuils de la zone A.

ZONE C : graduellement, jusqu'à la fin de la forme, la transparence interne augmente. En conséquence, l'évolution des seuils s'effectue progressivement de 50 à 62 .

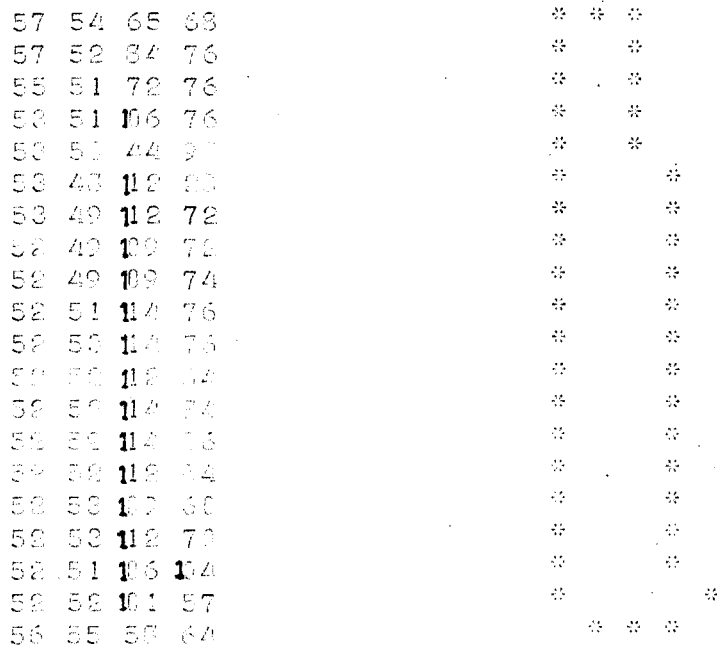


Figure III.9a: Contour d'une forme allongée.



Figure III.9b : Contour d'une forme compacte.

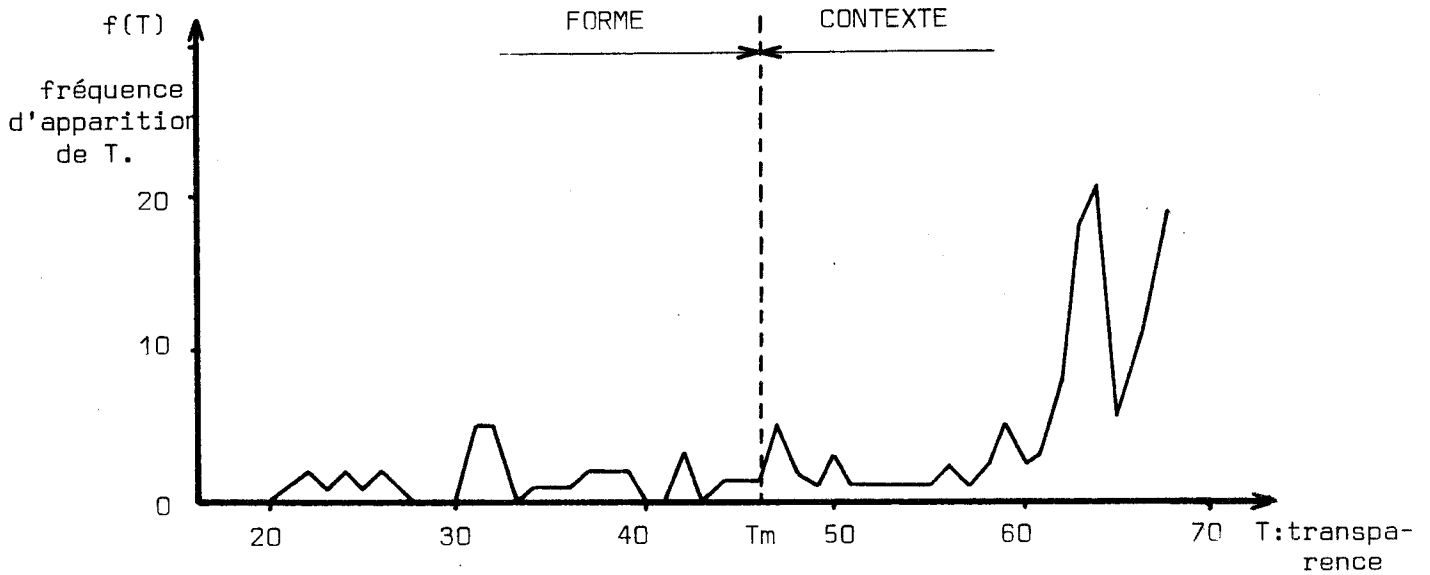


Figure 10.a : Histogramme pour une forme compacte

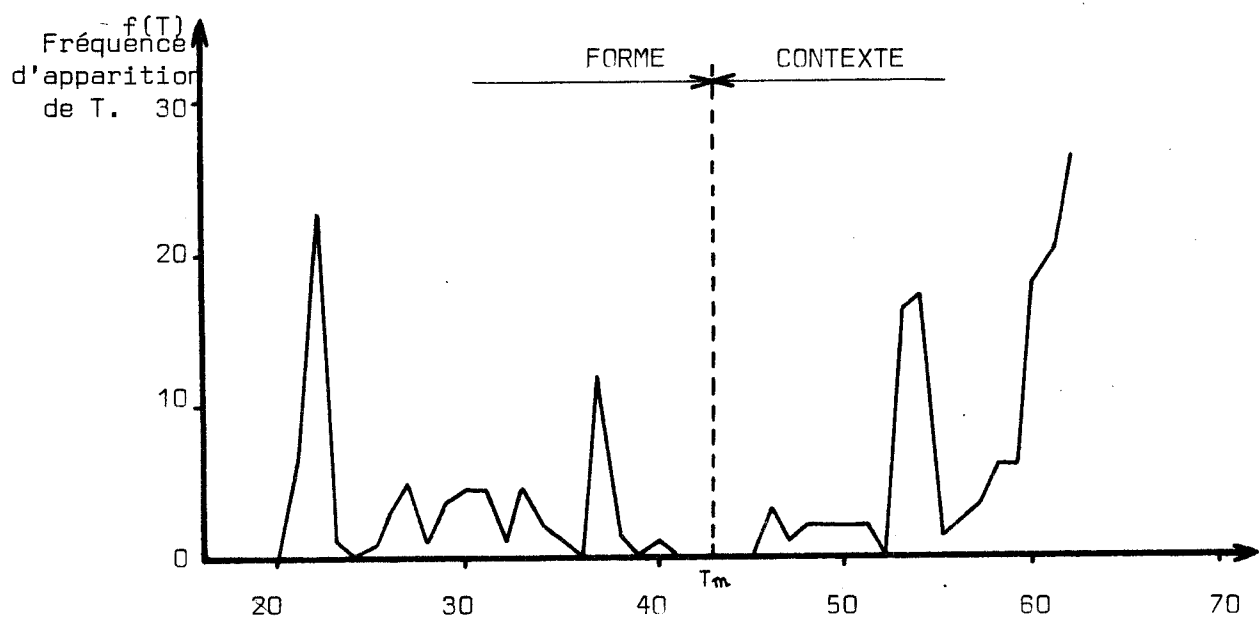


Figure 10.b : Histogramme pour une forme allongée.

T_m : Valeur moyenne des seuils S_g et S_d

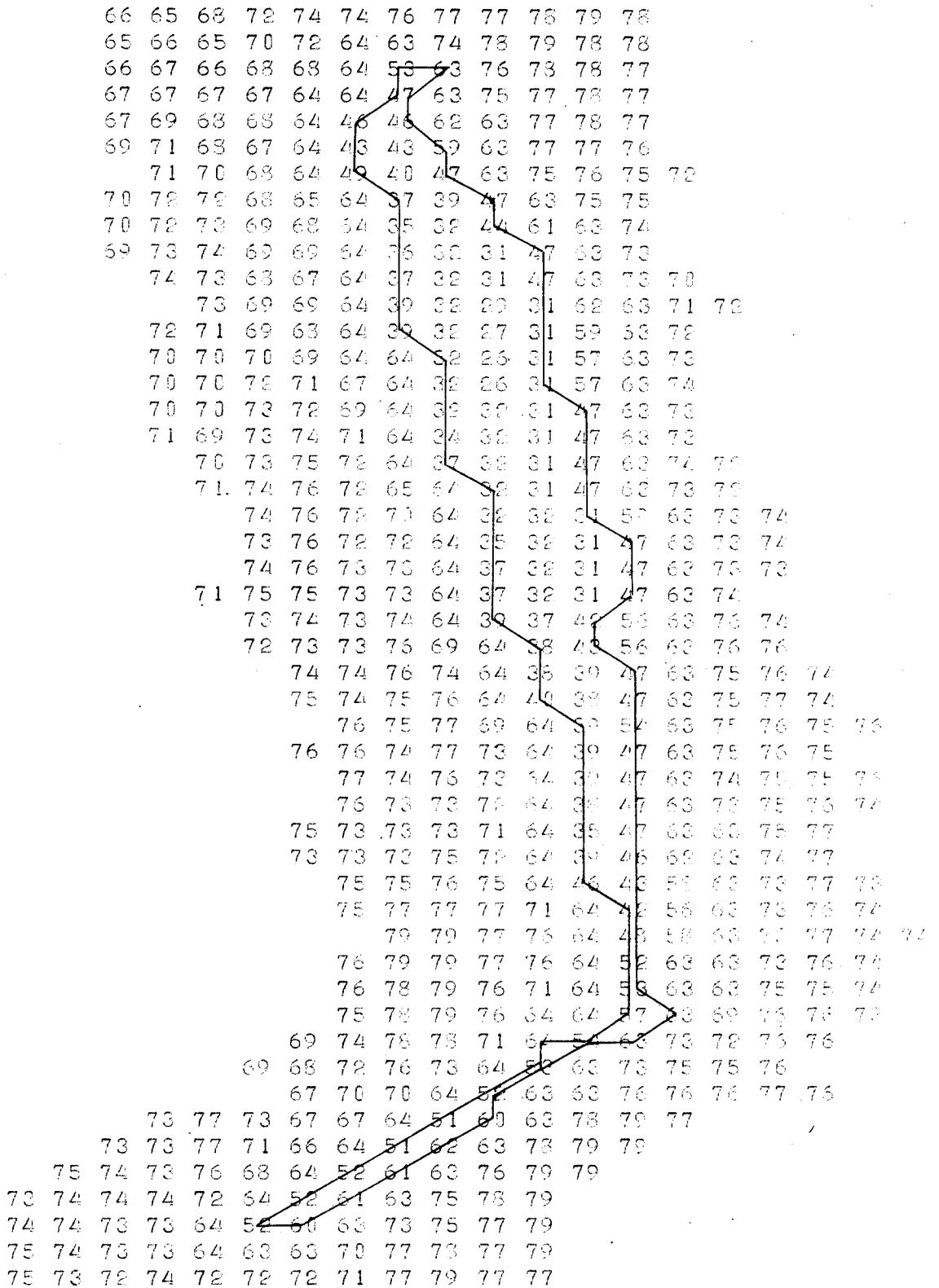


Figure III.11. : Sous-image d'une forme à transparence variable

S _g	S _d	P ₁	P ₂
58	64	88	92
56	62	36	112
55	61	44	82
55	61	44	90
53	59	96	93
53	59	96	96
53	59	96	84
54	55	76	112
51	53	128	112
50	52	98	106
47	51	98	109
47	50	86	94
46	50	52	96
48	51	48	112
50	51	48	112
51	51	106	114
51	51	106	112
51	51	52	112
52	51	48	112
52	51	109	112
52	51	109	114
54	53	40	78
55	55	101	76
55	56	44	96
56	56	44	98
57	56	101	96
56	56	36	96
56	56	36	93
55	55	36	93
54	55	44	80
54	55	32	70
57	56	116	80
58	56	93	82
60	58	112	66
62	60	96	56
63	62	61	58
64	64	50	50
65	63	64	76
64	63	80	80
62	63	72	64
59	63	104	72
61	63	52	72
62	63	64	64
62	63	80	61
62	62	84	56

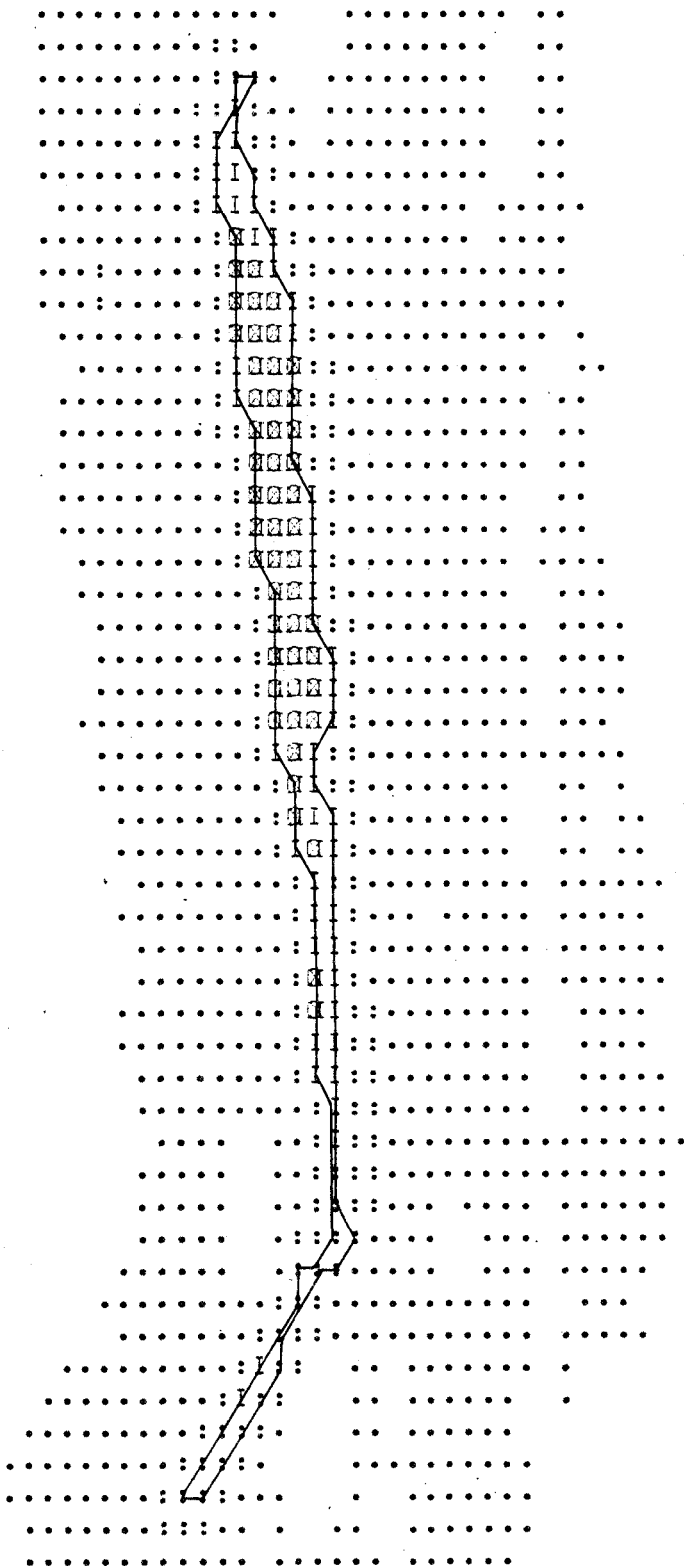


Figure III.12. : Dessin d'une forme à transparence variable

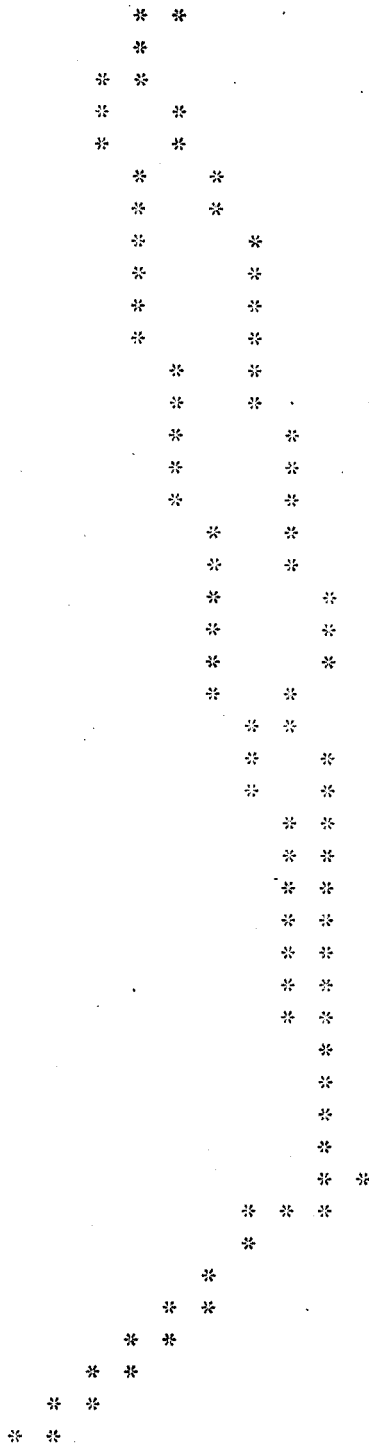


Figure III.13 : contour d'une forme à transparence variable

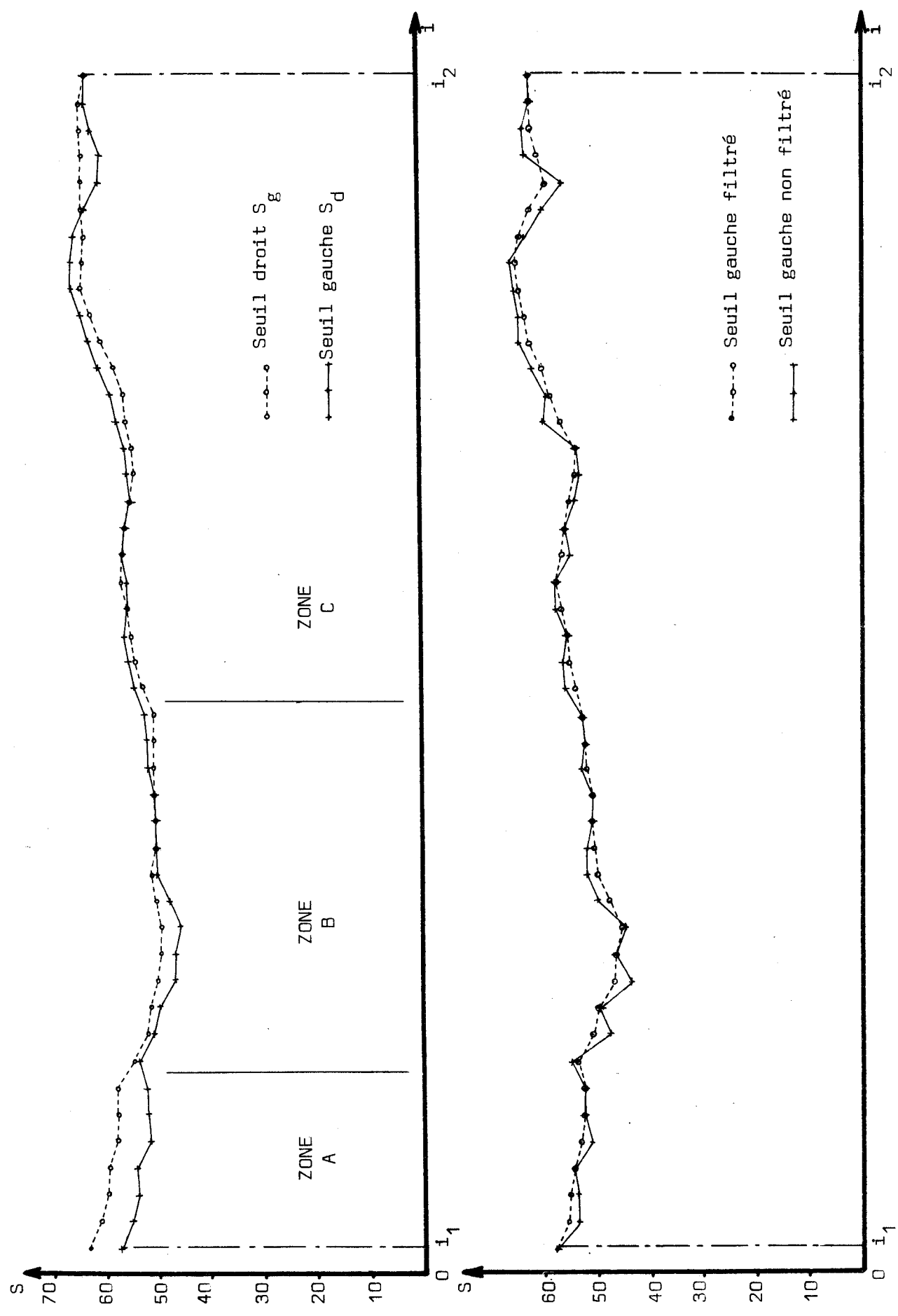


Figure II.14 : Evolution des seuils en fonction de l'indice de ligne

III.6. Conclusions

La méthode de recherche de contour exposée dans ce chapitre possède quelques propriétés qu'il est intéressant de résumer :

- le procédé permet une adaptation aux changements de transparence, même importants, du contexte et de la forme elle-même.
- cette méthode est très rapide, car seules des opérations simples sont utilisées (seuillage).
- la connaissance de la sous-image entière n'est pas nécessaire, car le traitement s'effectue ligne par ligne.

Nous verrons dans le chapitre VI. , que cette dernière propriété nous permet de réduire la mémoire destinée aux sous-images, par une acquisition et un traitement quasi-simultanés.

C H A P I T R E IV

- PARAMETRISATION DES FORMES

En vue de la reconnaissance des formes, s'appuyant sur l'analyse de certains paramètres géométriques, la première partie de ce chapitre est consacrée à la représentation des contours grâce au codage de Freeman.

Pour répondre au problème précis de reconnaissance de défauts dans une matière textile, seuls les éléments géométriques suivants sont recherchés :

- un paramètre représentant la compacité permettant de dissocier les formes allongées des formes compactes.

- la dimension maximale d'une forme permet le rejet de certaines formes qui ne sont pas dans les normes de prise en compte .

Le calcul de ce dernier paramètre a suscité le développement d'un algorithme rapide facilitant ainsi une reconnaissance des formes en temps réel.

IV.1.Représentation des contours de forme par le codage de Freeman{42,43}.

Nous avons vu, au chapitre III, que le contour d'une forme est défini par l'ensemble des coordonnées (X_1, Y_1) des points frontières. Cette définition n'est pas adaptée au calcul de certains paramètres géométriques utiles à la reconnaissance, tels que périmètre , longueur, etc.....

Freeman a introduit une représentation des contours de forme dont l'une des propriétés est de simplifier la recherche de ces paramètres.

IV.1.1 Connexité par "huit-voisinage".

La représentation des contours grâce au codage de Freeman nécessite au préalable la définition de "huit-voisinage". Deux points sont appelés "huit-voisins" si leur distance est égale à "a" ou " $\sqrt{2}a$ ", "a" étant le pas d'échantillonnage de l'image. La figure IV.1 montre qu'un point a huit possibilités de huit-voisinage.

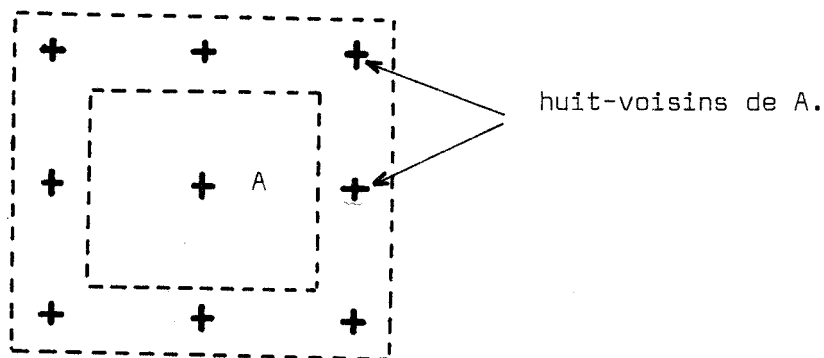


Figure IV.1 : Possibilités de huit-voisinage

Une ligne de contour est défini par un ensemble de N points $\{ P_i, i \in [1, N] \}$ tels que P_k et P_{k-1} soient huit-voisins. Une ligne de contour est fermée si d'autre part P_1 et P_n sont également huit-voisins.

IV.1.2. Représentation des contours grâce à des éléments directionnels

Freeman a remplacé l'ensemble des points de contour P_i par un ensemble $\{ u_i, i \in [1, N] \}$ où la valeur $u_i = f(P_i, P_{i-1})$. L'élément u_i est proportionnel à l'angle formé entre un axe de l'image (axe des lignes) et la direction définie par P_{i-1}, P_i . La figure IV.2. représente les huit valeurs possibles de u_i :

$$u_i \in [0, 7]$$

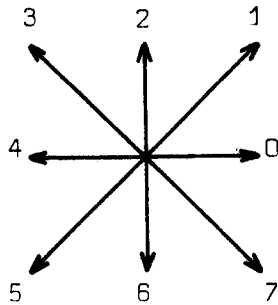


Figure IV.2. : Valeurs de u_i

L'ensemble formé par ces éléments u_i est appelé chaîne de directions. L'exemple de la figure IV.3. représente une forme dont le contour est défini par le codage de Freeman.

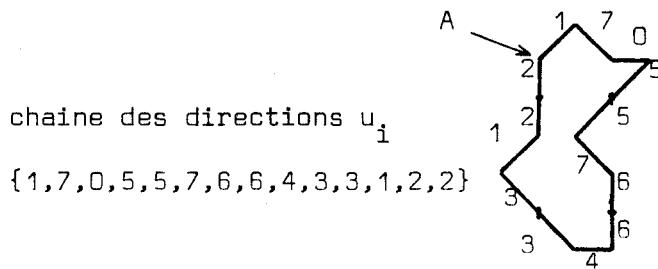


Figure IV.3.: Exemple de codage d'un contour

L'ensemble initial des points de contour P_i peut être retrouvé à partir de l'ensemble u_i et des coordonnées d'un point de la forme. Cette représentation permet donc une compression d'informations, car chaque couple de coordonnées définissant un point est remplacé par une valeur comprise en 0 et 7, donc pouvant être codée informatiquement sur 3 bits.

IV.1.3. Adaptation au mode séquentiel de recherche de contour

La recherche de contour s'effectue de façon séquentielle, c'est-à-dire ligne par ligne. Il est donc intéressant de constituer la chaîne de directions au fur et à mesure de la détection des points frontières. Pour cette raison, l'ensemble des directions élémentaires u_i se divise en deux parties : une chaîne de contour droit reliant tous les points frontières droits de la forme, et symétriquement une chaîne gauche.

Les contours étant préalablement définis par un ensemble de points frontières qui ne sont pas forcément "huit-voisins", il faut assurer la fermeture entre ces points : les positions C_g^i ou C_d^i sont reliées aux positions des points frontières de la ligne suivante C_g^{i+1} ou C_d^{i+1} . La liaison est effectuée, en choisissant arbitrairement l'un des plus courts chemins reliant deux points (figure IV.4).

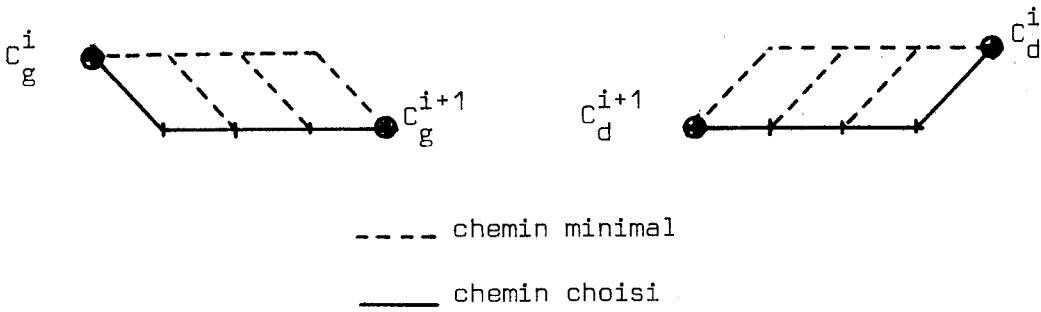
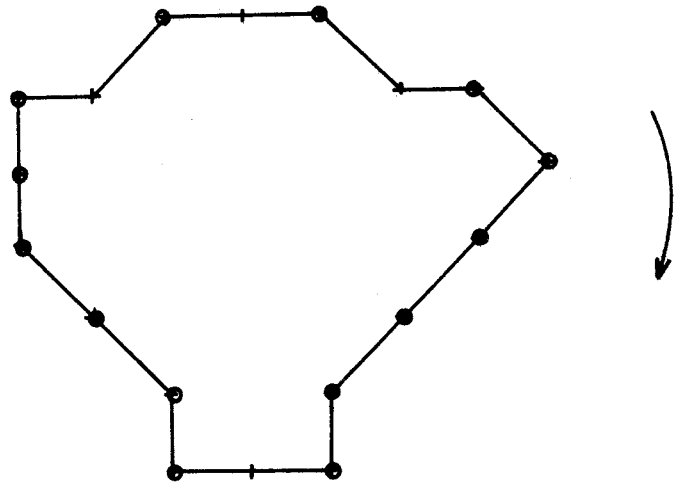


Figure IV.4.: Fermeture du contour

La figure IV.5 donne un exemple de passage de l'ensemble des points frontières à un contour fermé

contour gauche	contour droit
-	0,0
0,1	7,0
2	7
2	5
3	5
3	5
4,4,2	6



● : point frontière (C_g^i ou C_d^i)

+ : point de fermeture

chaîne totale : {0,0,7,0,7,5,5,5,6,4,4,2,3,3,2,2,0,1}

Figure IV.5.: Exemple de recherche d'une chaîne de contour

IV.2. Mesure de compacité

IV.2.1. Formulation de la compacité

Le concept de compacité d'une forme est utilisé comme paramètre séparateur des formes allongées des formes ovoïdes . Une première formulation de la compacité peut être envisagée par le rapport entre les dimensions extrémales des formes. Toutefois, une telle définition de la compacité n'est pas aisément calculable.

La compacité est alors représentée par le biais d'un autre indice sans dimension plus facilement mesurable : le rapport P^2/s , P et S étant respectivement le périmètre et la surface de la forme.

Si l'on examine les propriétés d'un tel indice, on remarque qu'il est :

- indépendant de la taille des formes examinées : deux formes homothétiques ont le même indice de compacité.

- séparateur au sens de la reconnaissance des formes. Ainsi, pour un cercle et un carré, formes de compacité voisine, l'indice vaut respectivement 4π et 16. Pour un rectangle, sa valeur tend vers le rapport longueur sur largeur ($4L/l$) comme le montre la figure IV.4.

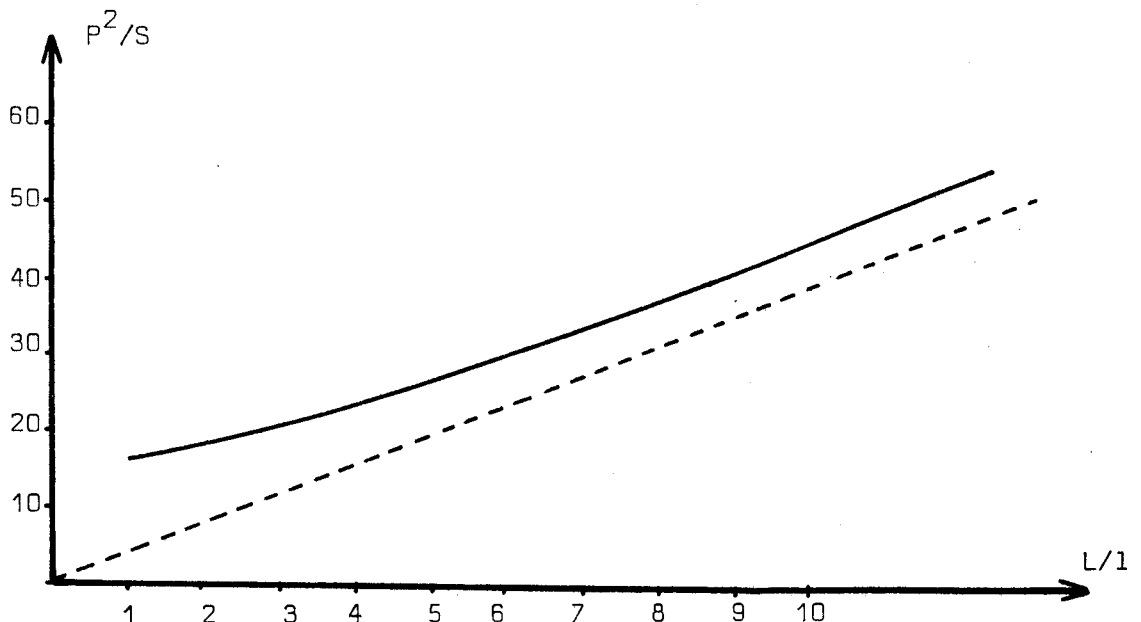


Figure IV.4.: Relation entre P^2/S et L/l

IV.2.2. Mesure de périmètre.

Un contour de forme est constitué d'une chaîne fermée d'éléments verticaux, horizontaux et diagonaux. Or, chaque élément vertical ou horizontal apporte au périmètre une contribution égale à "a" , pas d'échantillonnage de l'image; de plus, ces éléments sont facilement repérables car les codes, au sens de Freeman, sont pairs. De même, tout élément diagonal, dont l'imparité du code permet le repérage, contribue au périmètre par une valeur égale à " $a\sqrt{2}$ " .

Ainsi, la mesure de P est obtenue par la formule :

$$P = a \times (\text{nombre d'éléments de code pair} + \sqrt{2} \text{ nombre d'éléments de code impair}).$$

IV.2.3. Mesure de surface

L'algorithme de calcul de surface repose sur la procédure utilisée dans le cas d'une forme $f(x,y)$ continue en x et y.

Définissons un axe parallèle aux lignes de l'image à partir d'un point quelconque du contour, A, pris comme point de départ du calcul. La surface s'obtient par intégrations successives de parties de surface s_i comprises entre le contour et l'axe (figure IV.7). L'intégration s'effectue en parcourant tout le contour pour revenir au point de départ (A).

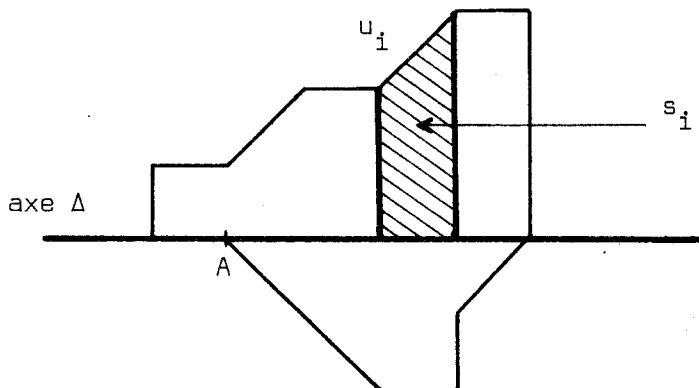


Figure IV.7.: Définition de l'élément de surface s_i

Soit un élément u_i de la chaîne de directions défini par les points P_{i-1} , P_i . Pour chaque pas de calcul on définit les variables suivantes :

$$D_i = D_{i-1} + d_i \text{ (avec } D_0 = 0 \text{ au point A)}$$

D_i est la distance du point P_i à l'axe . L'élément d_i est calculé simplement à partir de u_i (voir figure IV.8).

S_i est la portion de surface intégrée de P_0 à P_i . L'élément de surface s_i est calculé à partir de D_{i-1} et u_i :

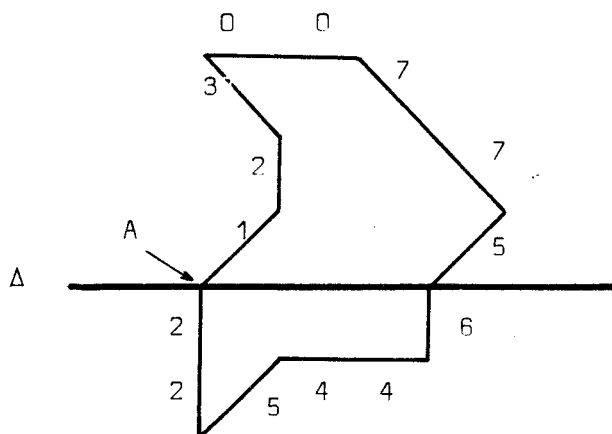
$$s_i = f_p(u_i) \cdot D_{i-1} + f_s(u_i)$$

Les valeurs $d_i, f_p(u_i), f_s(u_i)$ sont données en fonction de u_i dans le tableau de la figure IV.8.

u_i	d_i	$f_p(u_i)$	$f_s(u_i)$	s_i
0	0	1	0	$D_{i-1} + 1$
1	1	1	1/2	$D_{i-1} + 1/2$
2	1	0	0	0
3	1	-1	-1/2	$-D_{i-1} - 1/2$
4	0	-1	0	$-D_{i-1}$
5	-1	-1	1/2	$-D_{i-1} + 1/2$
6	-1	0	0	0
7	-1	1	-1/2	$D_{i-1} - 1/2$

Figure IV.8 : Relations entre $d_i, f_p(u_i), f_s(u_i), s_i$ et u_i

La figure IV.9 donne un exemple de calcul de surface dont le résultat final est $S=10$ éléments de surface $a \times a$.



u_i	D_i	s_i	S_i
1	1	$1/2$	$1/2$
2	2	0	$1/2$
3	3	$-2-1/2$	-2
0	3	3	1
0	3	3	4
7	2	$3-1/2$	$5+1/2$
7	1	$2-1/2$	7
5	0	$-1+1/2$	$6+1/2$
6	-1	0	$6+1/2$
4	-1	1	$7+1/2$
4	-1	1	$8+1/2$
5	-2	$1+1/2$	10
2	-1	0	10
2	0	0	10

Figure IV.9.: Exemple de calcul de surface

IV.3. Dimensionnement d'une forme

IV.3.1. Définition de la longueur d'une forme

La longueur d'une forme, en tant que plus grande dimension, répond à la définition suivante :

$$L = \max d (P_i, P_j); i, j \in [1, N]$$

- N : dimension de la chaîne de contour
- $d (P_i, P_j)$: distance entre P_i et P_j .

Le calcul systématique des distances de tous les couples (P_i, P_j) est exclu car il nécessite $N(N-1)/2$ distances à établir, ce qui est incompatible avec une recherche rapide, dans le cas où N est grand.

L'algorithme proposé, reposant sur un processus itératif peut être représenté par un organigramme (figure IV.10).

L'algorithme a pour principe, le déplacement de deux points A, B le long du contour de la forme. Le sens de déplacement, soit de A ou de B, est choisi de telle façon que la distance $d(A, B)$ augmente à chaque pas d'itération.

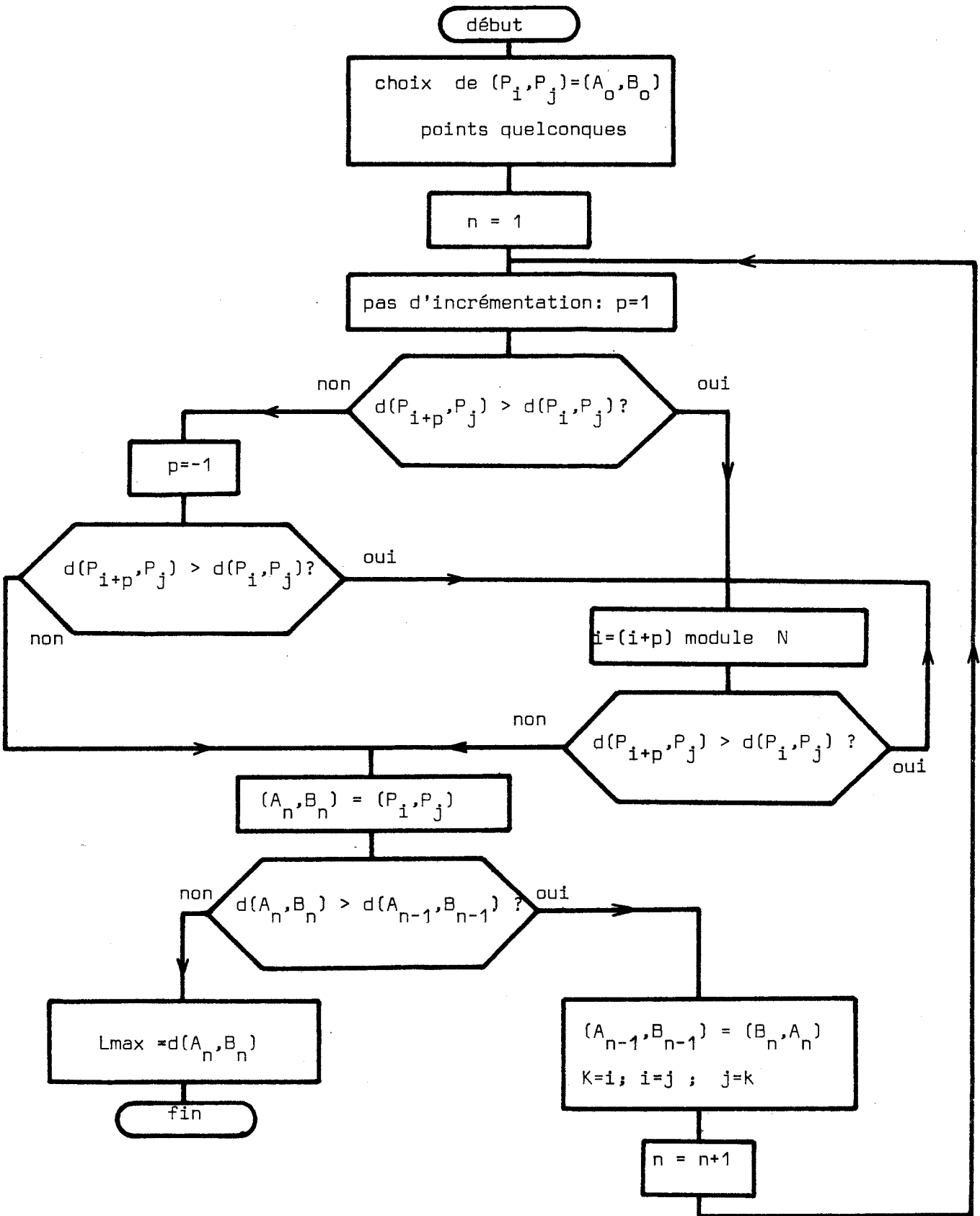


Figure IV.10 : Algorithme de calcul de longueur.

La figure V.11 montre un exemple d'emploi de cet algorithme.

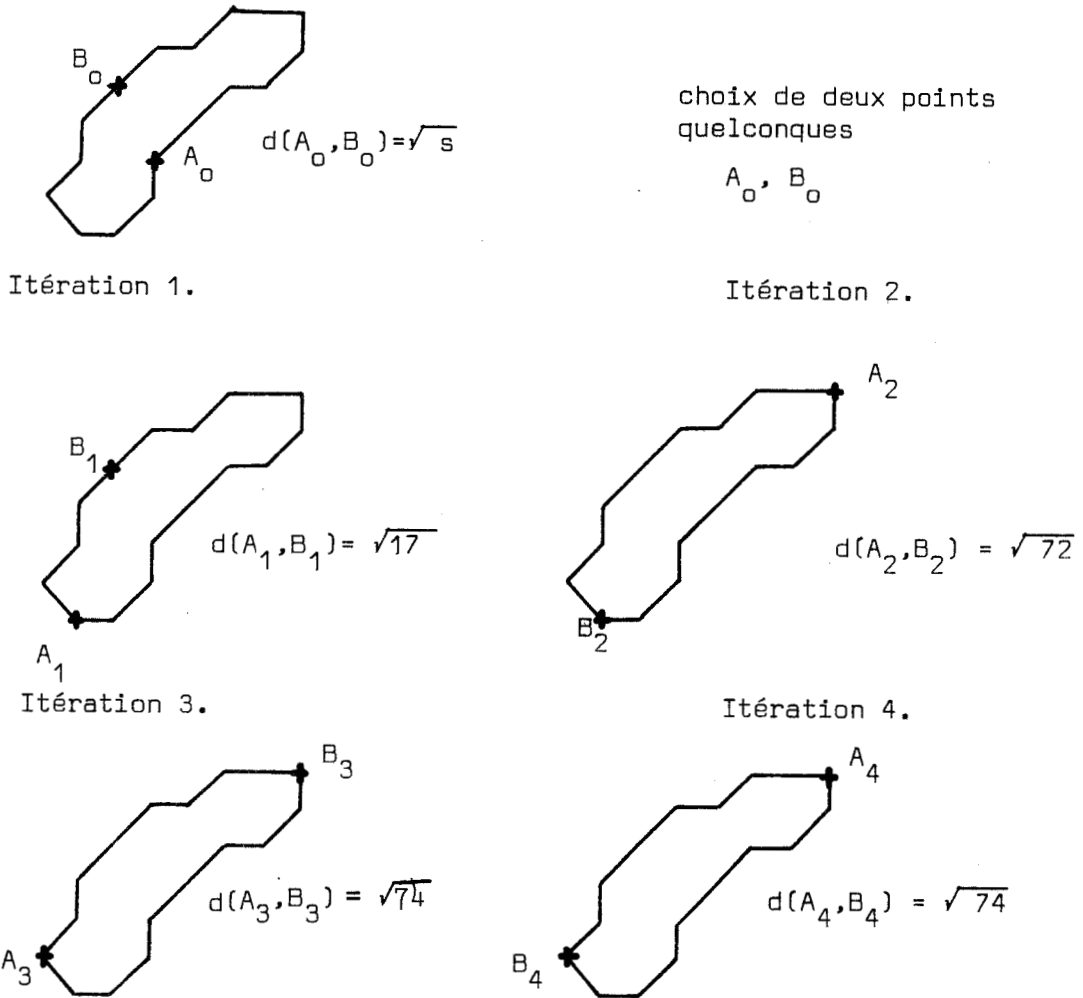


Figure IV.11. Illustration de la procédure

Le gain de temps obtenu grâce à ce procédé est appréciable; toutefois un guidage du choix initial de P_i, P_j permet d'améliorer notablement la rapidité de la méthode. Par exemple, P_i et P_j sont les points de contact du rectangle exinscrits de côtés parallèles à l'axe de l'image. Pour l'exemple de la figure IV.11, le choix de ces points initiaux a permis de ramener le calcul de longueur à 2 itérations.

IV.3.2. Optimalité du résultat

Il faut toutefois craindre que la méthode précédemment décrite fournisse un maximum local comme l'illustre la figure IV.12

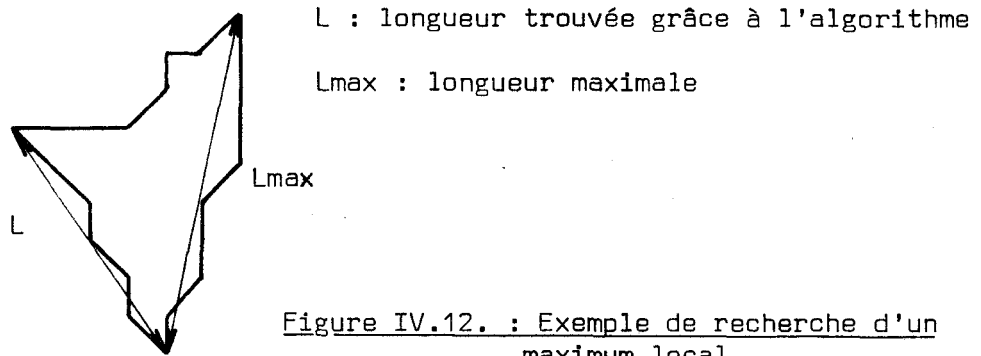


Figure IV.12. : Exemple de recherche d'un maximum local

Pour éviter les erreurs dues au maxima locaux, l'algorithme est suivi d'une phase supplémentaire :

- calcul de $d(A, P_k)$ et $d(B, P_k)$ $k \in [1, N]$

A et B étant les points trouvés après l'emploi de l'algorithme précédemment défini :

- Si l'une des distances $d(A, P_k)$ ou $d(B, P_k)$ est supérieure à $d(A, B)$, l'algorithme complet est repris, en considérant les couples (A, P_k) ou (B, P_k) comme point de départ.

La forme b de la figure IV.13 montre qu'une telle extension de l'algorithme ne conduit pas toujours au résultat escompté : la distance $d(C, D)$, longueur maximale de la forme, n'est pas détectée .

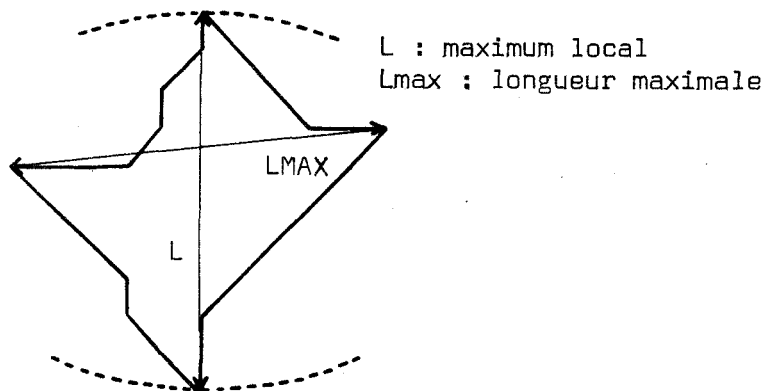


Figure V.13 : Cas où la longueur maximale n'est pas trouvée grâce à l'extention de l'algorithme.

En fait, un tel cas apparait lorsque la forme est compacte. Par conséquent, lorsque la compacité P^2/s est inférieure à un seuil séparant les formes compactes et allongées, la définition, $L = \max (P_i, P_j), i \text{ et } j \in \{1, N\}$ est employée. Le temps requis par l'emploi de cette définition reste faible car, dans notre application, les formes compactes ont des dimensions très faibles.

CHAPITRE V

REALISATION DU PROTOTYPE

Après avoir présenté le problème textile de détection de défauts dans les peignés de laine, ce chapitre aborde la réalisation matérielle du procédé original d'acquisition et de traitement d'images développé dans les trois précédents chapitres. Ce procédé a permis l'implantation d'un matériel informatique qui, jusqu'à présent, était peu destiné au traitement d'images : le microcalculateur. Les performances de l'ensemble caméra calculateur ont été accrues en implantant un système à accès direct mémoire dont le rôle est d'acquérir des parties d'images en ralentissant au minimum le microprocesseur. Pour cela les possibilités de travail du microcalculateur en multi-processeur ont été exploitées.

V.1. Présentation du problème textile

L'industrie textile est constamment confrontée aux problèmes de contrôle de qualité, soit des produits finis (tissus) soit des produits en cours de fabrication. Certains contrôles ont encore, actuellement, pour base une inspection visuelle généralement empreinte de la subjectivité des opérateurs. Notamment, dans l'industrie lainière, l'examen visuel des rubans de peigné conduit à des résultats peu satisfaisants.

Après avoir été lavée, la laine, à l'état brut, est cardée :
- cette opération a pour rôle d'homogénéiser la répartition des fibres.
Avant d'être envoyée en filature, la matière est peignée : toutes les fibres sont ainsi parallélisées. A ce niveau, un contrôle de qualité de la laine, présentée en rubans de peigné, permet d'estimer la qualité du produit fini.

Les résultats sont importants, du point de vue commercial, car ils sont une base de discussion du prix de la laine, vendue par les peigneurs aux filateurs.

La "note" attribuée à la laine peignée tient compte de plusieurs composantes :

- finesse et longueur des fibres : ces paramètres permettent d'estimer la qualité et la solidité du tissu.

- propreté de la laine

C'est en particulier, ce dernier point qui pose des problèmes aux laboratoires d'analyse textile chargés de ce contrôle. L'examen de la propreté des peignés consiste à prélever un échantillon de poids donné (100 gr), et à compter les impuretés présentes dans le ruban de peigné. Il existe deux types d'impuretés :

- matières végétales : certaines particules telles que morceaux de paille, chardon n'ont pas été éliminées par les différents traitements effectués sur la laine. Les pailles de longueur inférieure à 3 mm ne sont pas comptabilisées, car les impuretés végétales de petite taille sont éliminées lors de la filature ou n'entraînent que très peu de conséquences au niveau du tissu.

- agglomérats de fibres : des amas de fibres, appelés boutons ou bouchons selon leur taille, provoquent un aspect "pelucheux" sur le produit fini. En outre, ces impuretés occasionnent en filature la cassure des fils. Seuls les boutons de diamètre supérieur à 1 mm sont pris en compte.

Actuellement, le contrôle visuel d'échantillons de peigné consiste à étirer le ruban de façon à présenter devant l'opérateur un fin voile de fibres, dans lequel il est possible de discerner les impuretés. La détection visuelle des défauts s'effectue généralement avec facilité, bien que certaines largeurs de paille soient faibles (0,1 mm). Par contre les dimensions des boutons sont actuellement plus difficiles à appréhender par les opérateurs, car leurs contours ne sont pas nets, conduisant à des résultats variables d'un opérateur à l'autre.

Des essais comparatifs inter-laboratoires ont mis en évidence la subjectivité des résultats. Des études ont porté sur des boutons aux dimensions très proches des normes de rejet (1 mm). Les résultats, provenant de plusieurs opérateurs, ont variés jusqu'à un rapport de 1 à 10.

Un appareil d'inspection automatique est donc d'un intérêt capital non seulement pour les laboratoires d'analyse textile, mais aussi pour les peigneurs. L'examen de propreté de la laine leur permettrait éventuellement de modifier certains paramètres de leurs machines de façon à éliminer certains défauts.

Ce dernier marché potentiel constitué par les industries textiles, nous a amené à choisir un matériel (capteur - calculateur) peu coûteux, et donc à développer des méthodes qui peuvent s'adapter à ce type de matériel.

V.2. Présentation du capteur optique

V.2.1. Dispositif de présentation de la laine

Il a été nécessaire d'améliorer la présentation du voile de laine, car le dispositif actuellement utilisé pour une détection visuelle présente quelques inconvénients. En particulier, des imperfections de l'étirage peuvent contraindre les opérateurs à arrêter le dispositif pour une analyse visuelle plus fine du voile.

Actuellement, la matière textile est étirée entre deux rouleaux dont les vitesses fixent le rapport d'étirage. Pour obtenir un étirage correct, l'écartement entre les rouleaux doit être supérieur aux plus longues fibres du peigné. Cette condition est réalisée au détriment des fibres courtes qui sont très peu maintenues. Une meilleure homogénéité du voile de laine est obtenue en insérant deux rouleaux réunis par une bande en caoutchouc dont le rôle est de faciliter le cheminement des fibres courtes (figure V.1).

Après étirage, le voile de laine est légèrement tendu, grâce à un rouleau dont la vitesse est faiblement supérieure à celle du rouleau étireur. La prise de vue s'effectue par transparence, entre ces deux rouleaux.

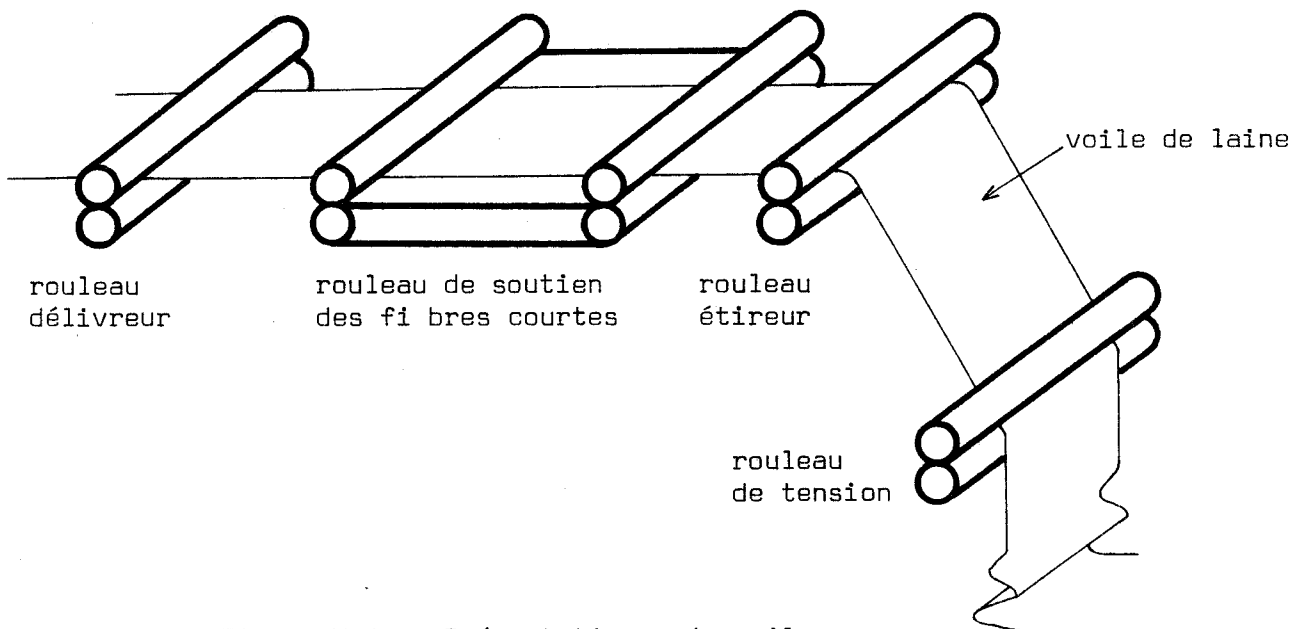


Figure V.1. : Présentation du voile de laine

V.2.2. Choix et description du capteur

Un capteur mono-dimensionnel convient à notre application car la deuxième dimension de l'image s'obtient par le déplacement continu de la laine. Parmi les capteurs ayant cette caractéristique, nous pouvons écarter les systèmes utilisant un tube flying spot ou à balayage d'un faisceau lumineux grâce à un miroir tournant, car leur coût est relativement élevé.

Notre choix s'est arrêté sur les réseaux linéaires de photodiodes intégrées.

La largeur totale à analyser (200 mm), le pas d'échantillonnage de l'image (0,2 mm) nous ont amené à utiliser un réseau linéaire de 1024 photodiodes. Ce capteur se présente sous la forme d'un circuit intégré, chaque élément photosensible ayant une surface de $0,25\mu \times 0,25\mu$. Comme il est impossible matériellement de disposer des 1024 signaux en parallèle, l'acquisition des données s'effectue séquentiellement. Dans ce but, le constructeur a intégré dans le circuit contenant les photodiodes, un registre à décalage de 1024 positions. Chacune des sorties du registre à décalage ouvre un interrupteur analogique MOS dont le rôle est d'aiguiller le signal de sortie d'une photodiode vers une sortie commune (figure V.2). Une horloge externe au circuit fait circuler une impulsion dans le registre à décalage, permettant ainsi un prélèvement séquentiel de toutes les informations. Une photodiode peut être considérée comme un intégrateur de lumière. Par conséquent le signal analogique de sortie est proportionnel à la quantité de lumière reçue entre deux prélèvements. La sortie du capteur est un signal vidéo par analogie au signal de sortie d'une caméra classique de télévision.

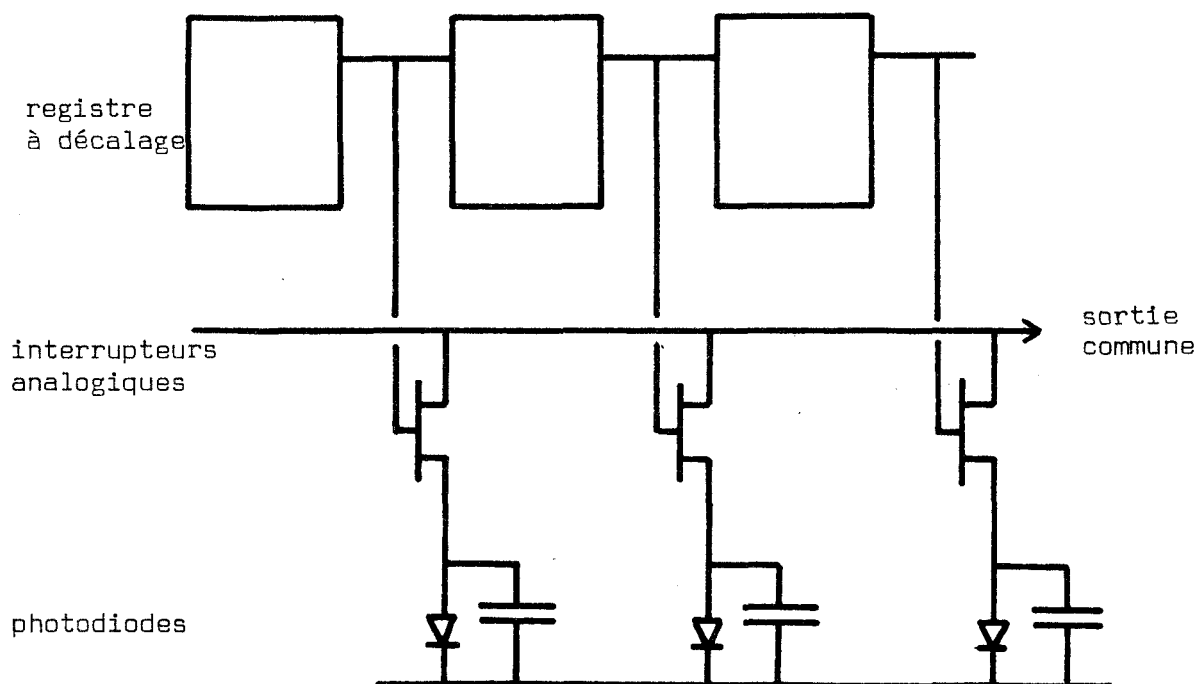


Figure V.2. : Présentation du capteur.

V.2.3. Optique et éclairage

La ligne de photodiodes est intégrée sur une longueur de 2,5 cm, alors que la largeur du peigné analysée est égale à 20 cm. Cette contrainte nous amène à intercaler un dispositif de focalisation de la lumière, entre le capteur et le peigné. Pour diminuer les aberrations optiques apportées par un objectif, le capteur est éloigné de la laine d'environ 50 cm, ce qui réduit l'angle de prise de vue à 22°.

Pour que l'analyse du peigné ne soit pas sensible à l'éclairage ambiant, une source lumineuse suffisamment puissante est placée derrière la matière textile (figure V.1). La perception s'effectue donc par transparence. La détection des impuretés se réalisant par seuillage adaptatif, une régularité parfaite de l'éclairage sur toute la largeur du peigné n'est pas nécessaire. Nous utilisons, tout simplement, une lampe à filament rectiligne. Par contre, la régularité de luminosité dans le temps est nécessaire. La lampe est donc alimentée par une source de tension continue.

V.3. L'unité de traitement informatique

V.3.1. Choix d'un calculateur

Notre principal critère étant une recherche de moindre coût, nous nous sommes naturellement tournés vers un matériel informatique à base de microprocesseur. Ce matériel est généralement adapté au contrôle ou à la commande de processus lents. Il est communément admis que le traitement d'images, appelé à manipuler un volume important de données, requiert un matériel puissant organisé autour de périphériques rapides. Cependant, nous avons défini une procédure d'acquisition et de traitement d'images dont le but est de diminuer les exigences matérielles.

Le microprocesseur choisi (INTEL 8080) est une unité centrale dont la base de travail est l'octet (8 bits).

Sa capacité d'adressage de 64.k octets est largement suffisante pour le stockage des programmes, images, sous-images et données intermédiaires de travail. La disparité et le nombre d'instructions alliés à la vitesse d'exécution confèrent à ce microprocesseur une puissance de programmation suffisante pour notre application.

V.3.2. Principales caractéristiques du micro-calculateur

La figure V.3 regroupe les principales caractéristiques du micro-calculateur bâti autour de l'unité centrale 8080 :

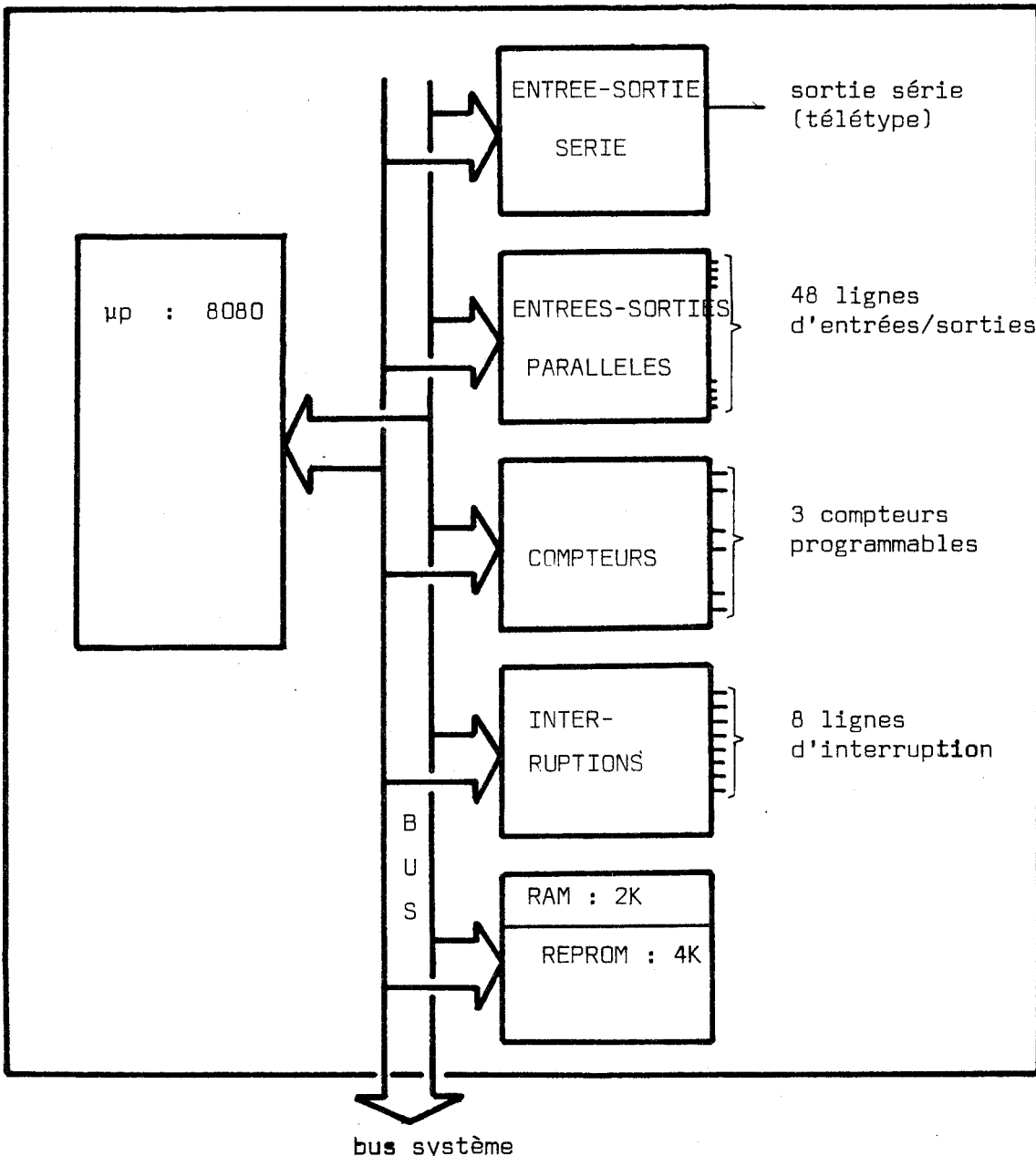


Figure V.3 : Structure générale du micro-calculateur

- interruptions vectorisées

Cette possibilité destine le micro-calculateur aux applications temps réel. Les interruptions permettent l'arrêt momentané d'une procédure par des événements extérieurs pour effectuer des tâches prioritaires.

- compteurs programmables.

Le micro-calculateur contient trois compteurs dont la valeur maximale de comptage peut être chargée par une opération logicielle. Hormis la génération d'horloge temps réel, l'emploi de ce matériel présente un autre intérêt : grâce à une opération logicielle, l'état des compteurs peut être acquis à n'importe quel moment du comptage.

- entrées-sorties parallèles et séries

Les entrées-sorties permettent la communication entre le microprocesseur et les périphériques. L'entrée-sortie série est connectée à une télécype. Les entrées et sorties parallèles ne sont pas employées pour la mise au point du prototype, mais peuvent être utilisées pour un éventuel affichage de résultats.

- Possibilités de travail simultané de plusieurs processeurs
(multi-processing).

Il est possible de connecter plusieurs micro-calculateurs en parallèle. Cette configuration nécessite une mémoire et un bus communs destinés à la communication entre les processeurs. Chaque calculateur peut travailler dans sa mémoire interne indépendamment des autres calculateurs. Le seul cas, nécessitant une gestion des priorités, apparaît quand plusieurs processeurs veulent échanger simultanément des informations avec la mémoire extérieure (figure V.4).

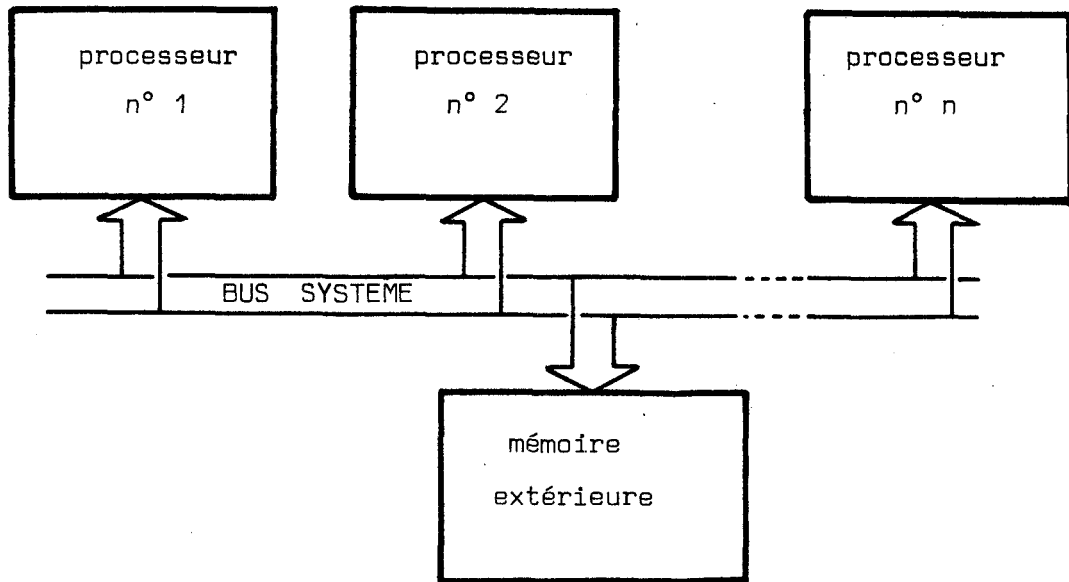


Figure V.4. : Connexion de plusieurs processeurs

V.4. Interface caméra - ordinateur

La figure V.5 représente l'interface entre l'unité de traitement et la ligne de photodiodes.

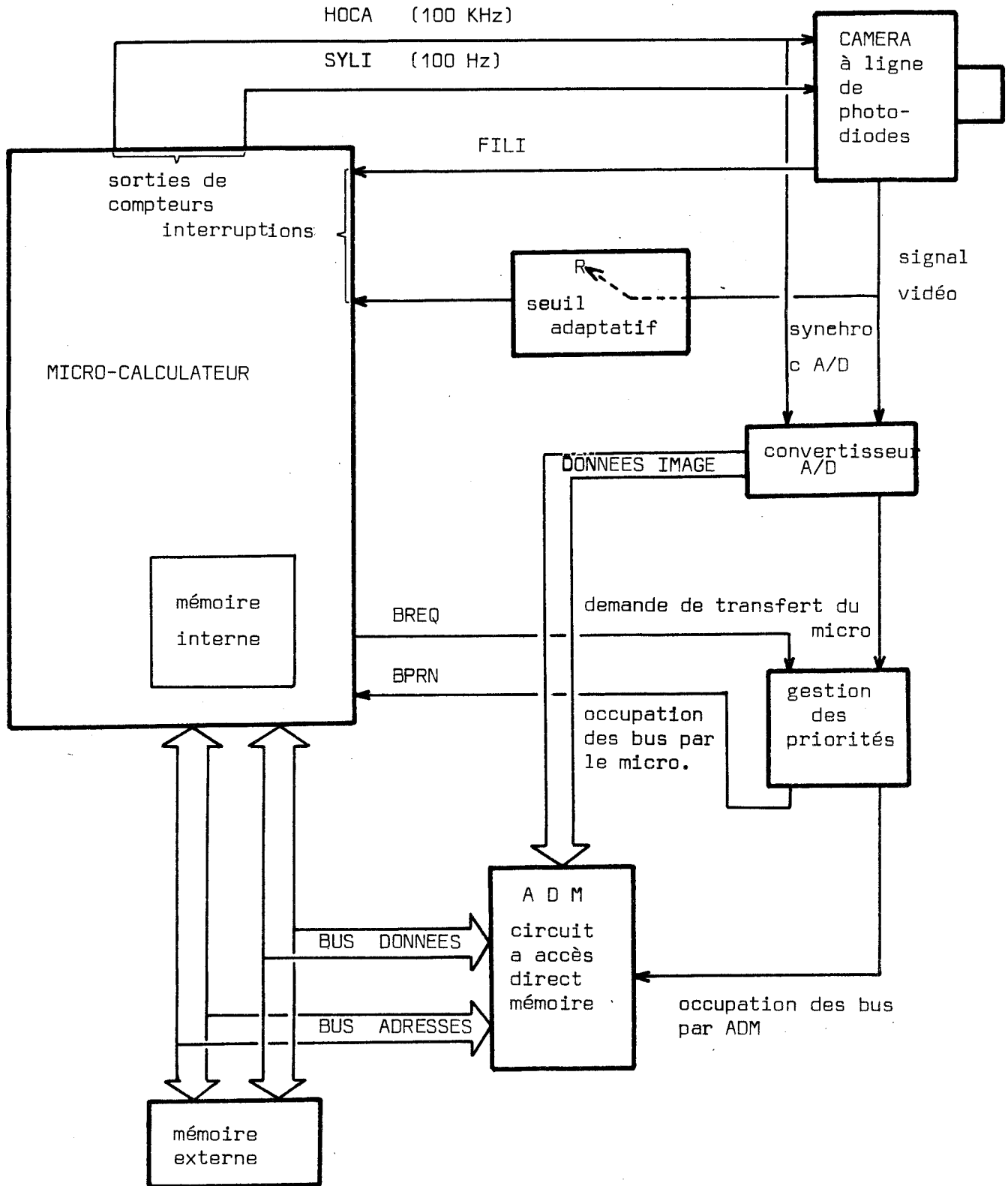


Figure V.5: Interface caméra microcalculateur.

V.4.1. Synchronisation de la caméra

La ligne de photodiodes associée à son électronique peut fonctionner de façon autonome. Cette propriété est notamment nécessaire pour des applications simples pour lesquelles la présence d'un ordinateur n'est pas nécessaire (détection de position). C'est pourquoi, d'une part le nombre de photodiodes scrutées peut être sélectionné (de 1 à 1024) par mini-interrupteurs. De même, le signal horloge (HOCA) peut être fourni par l'électronique de commande de la caméra. La période des impulsions HOCA multipliée par le nombre de photodiodes sélectionnées correspond au temps d'intégration de la lumière entre deux analyses d'une même photodiode. Le début de l'analyse de la première photodiode est provoqué par une impulsion de synchronisation ligne (SYLI).

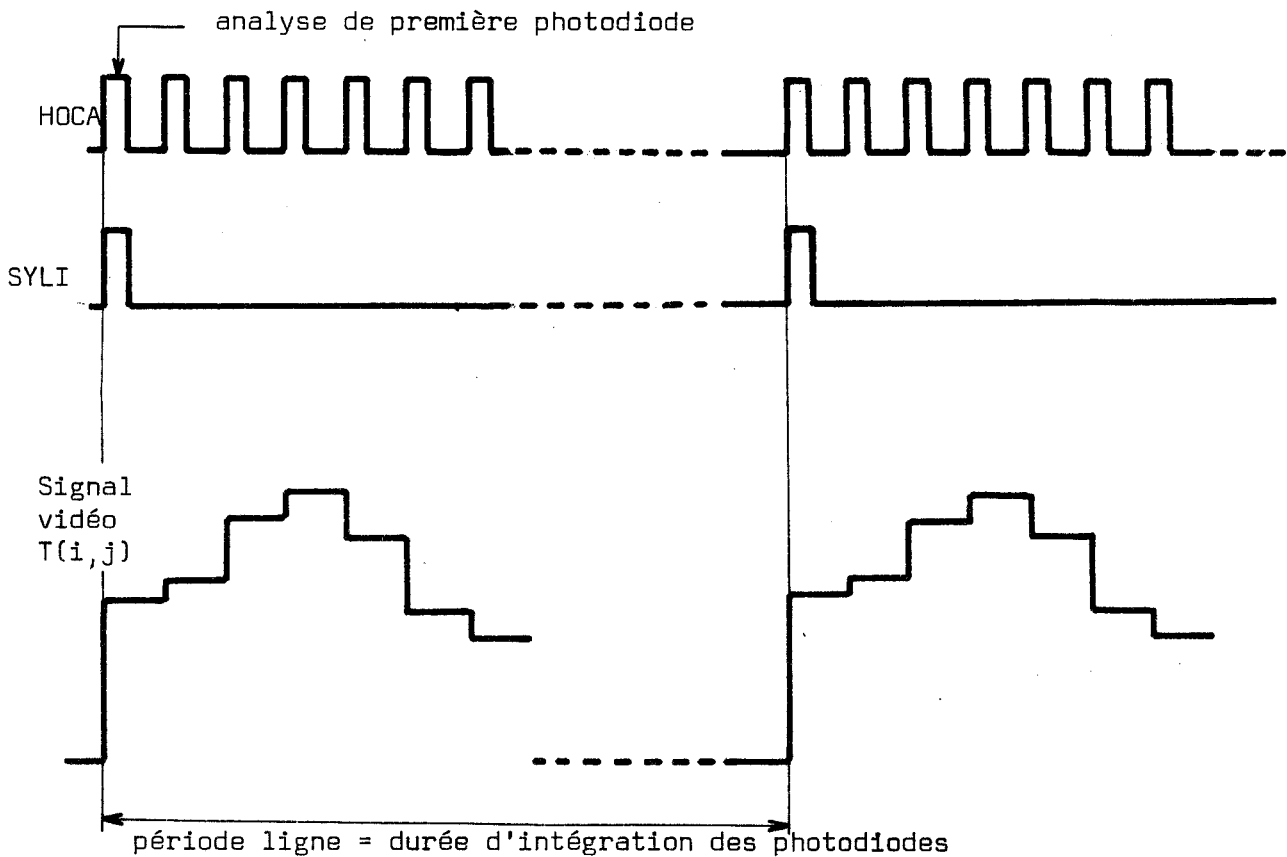


Figure V.6 : Signaux de synchronisation de la caméra.

Dans notre application, les signaux de commande de la ligne de photodiodes (HOCA et SYLI) sont générés par le micro-calculateur :

- HOCA est fourni par un compteur programmable. L'entrée du compteur est l'horloge à quartz, parfaitement stable qui pilote le micro-processeur. La période de HOCA est modifiable par voie logicielle.

- SYLI est fourni par un deuxième compteur programmable dont le signal d'horloge est HOCA . Le nombre de photodiodes analysées est donc fixé par logiciel.

La génération de SYLI par le biais d'un compteur programmable présente un autre intérêt : l'état du compteur peut être acquis à n'importe quel moment du comptage par une opération logicielle. Par conséquent, la position j_D^i (§ II.3.2) peut être acquise très simplement : il suffit de relever l'état du compteur lorsqu'une impulsion de détection de défaut survient.

Ce mode d'acquisition des positions j_D^i présente cependant l'inconvénient de nécessiter, au préalable, l'arrêt momentané du comptage. La période d'analyse de la ligne est ainsi augmentée de ce temps d'arrêt. Tout se passe donc, comme si le pas d'échantillonnage de l'image transversal à la ligne, est augmenté, entraînant par conséquent une distorsion dans l'image. Les répercussions de cet arrêt sont infimes car chaque lecture d'une position j_D^i n'entraîne qu'une augmentation de 1,2/1000 du pas transversal.

V.4.2. Interruptions

- interruption aléatoire

Lorsqu'une zone contrastée de l'image est détectée, le micro-calculateur est averti par le biais d'une interruption matérielle.

Le programme en cours est donc arrêté momentanément, pour permettre l'acquisition de la position d'une éventuelle forme (i, j_d^i) . Une ligne d'interruption du micro-calculateur est donc réservée au signal de détection de présence de formes.

- Interruption cyclique

Une interruption cyclique, dont la période est synchronisée sur le signal SYLI à les fonctions suivantes :

* incrémentement de l'indice de ligne i :

ce qui permet de constituer la structure bi-dimensionnelle de l'image.

* gestion des interruptions logicielles

Nous verrons au chapitre VI la fonction des interruptions logicielles. Nous pouvons seulement indiquer qu'un contrôle cyclique des tâches d'acquisition et de traitement des images est nécessaire. La période du cycle a été choisie égale au temps d'analyse d'une ligne, pour coupler à cette gestion temps réel, l'incrémentement de l'indice de ligne i.

Pour des raisons liées à l'acquisition des données (S'VI), le signal d'interruption n'est pas la synchro ligne SYLI, mais un signal généré par la caméra : FILI. Lorsque la période ligne est choisie supérieure à 1024 périodes de l'horloge HOCA, l'électronique associée à la caméra génère un signal indiquant la fin de scrutation des 1024 photodiodes : FILI (figure V.7)

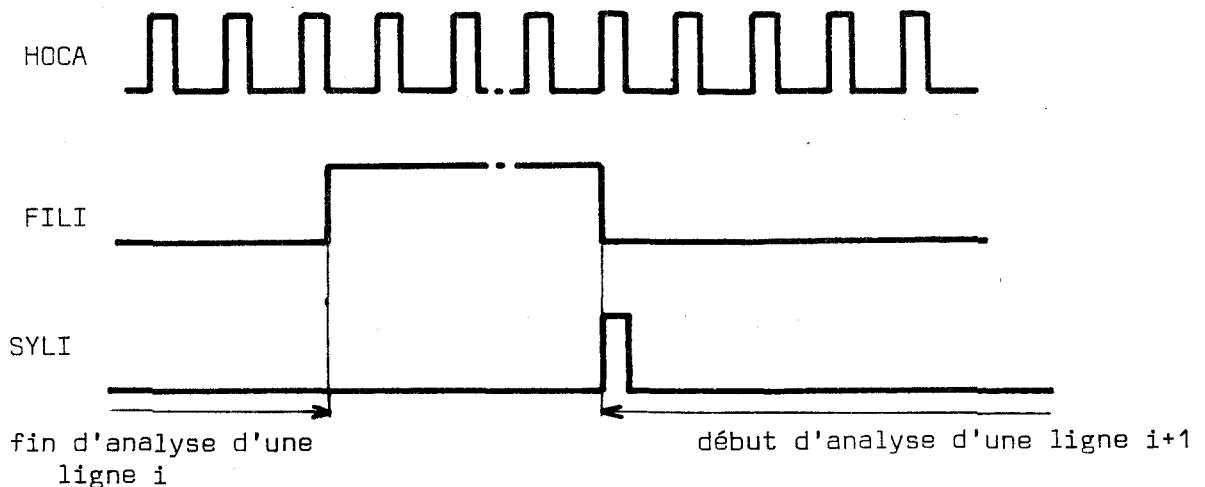


Figure V.7 : Indicateur de fin de ligne : FILI

V.5. Acquisition d'image fugitive par accès direct mémoire.

V.5.1. Organisation de la mémoire destinée aux images

La digitalisation du signal vidéo s'effectue grâce à une conversion analogique-digitale sur 8 bits. En fait un tiers de la dynamique du convertisseur est employé. L'échelle de gris est divisée en 80 niveaux, ce qui donne une définition amplement suffisante pour un traitement d'image.

Nous avons vu dans un chapitre précédent (§ II.4.2) que les données $T(i,j)$ des lignes d'indice $i-2, i-1$ doivent rester mémorisées pendant l'acquisition de la ligne i . Dans le chapitre suivant, nous verrons que, pour des raisons liées au traitement en temps réel, il est même nécessaire de connaître les données des lignes antérieures à $i-2$. Soit k , le nombre total de lignes d'informations mémorisées en permanence $(i-k, \dots, i)$

La figure V.8 illustre le mode de remplissage de la mémoire de $1024 \times k$ mots. Sur cette figure, chacun des segments schématise un secteur de $1k$ mots. Cette mémoire est considérée comme une pile, telle qu'à l'instant où la donnée $T(i,j)$ est acquise, l'information $T(i-k, j+1)$ se trouve au fond de cette pile.

L'image contenue dans cette mémoire étant continuellement renouvelée est appelée image fugitive.

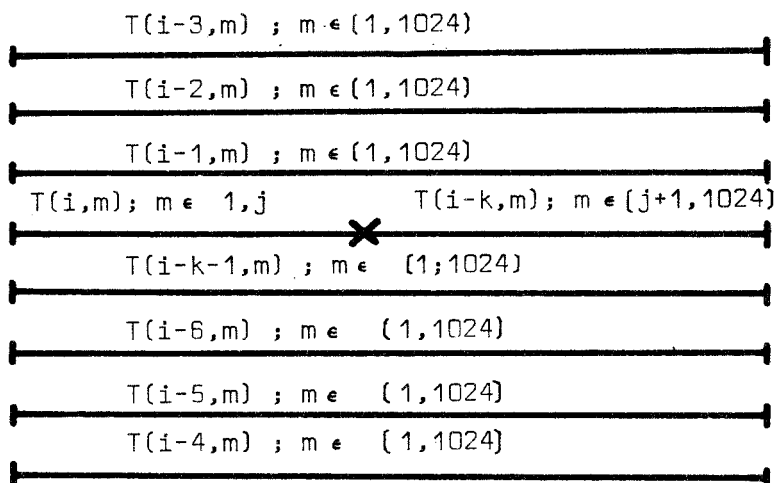


Figure V.8 :
Constitution de
l'image fugitive

La fréquence d'acquisition des valeurs $T(i,j)$ est de l'ordre de 100 kHz. Une telle fréquence est incompatible avec un rangement en mémoire des données par l'intermédiaire du microprocesseur. En effet, un programme d'interruption serait nécessaire pour synchroniser chaque acquisition. Le temps requis pour un transfert serait de l'ordre de 30 μ s.

Même si cette éventualité était matériellement envisagée, elle réduirait considérablement les performances du prototype.

Pour cette opération d'acquisition, un processeur spécialisé est donc utilisé, dont le rôle est de gérer les transferts des données dans la mémoire sans l'aide du microprocesseur. Cette fonction, généralement appelée accès direct mémoire (ADM), est précisée au paragraphe suivant.

V.5.2. Système à accès direct mémoire

L'accès direct mémoire est généré par un seul circuit dont on peut résumer le fonctionnement comme suit :

- Initialisation par le microprocesseur.

La première phase consiste à communiquer au circuit ADM , par voie logicielle :

- * l'adresse initiale de la mémoire où les données seront rangées.
- * le nombre total de transferts.

- Au rythme d'une horloge extérieure et sans intervention du microprocesseur les données digitales et les adresses de rangement en mémoire sont envoyées séquentiellement sur les bus données et adresses du micro-calculateur.

Un transfert par accès direct à la mémoire ne peut s'effectuer sans avoir l'assurance que les bus ne sont pas occupés par le microprocesseur. Avant chaque transfert une logique associée à l'ADM interroge l'unité centrale sur l'occupation des bus. Cette interrogation est réalisée au moyen de deux signaux :

- HRQ : activé par une demande d'occupation du bus par l'ADM.
- HLDA: activé par le microprocesseur lorsque le bus est libre.

La figure V.9 donne un exemple de liaison du circuit ADM au microprocesseur 8080.

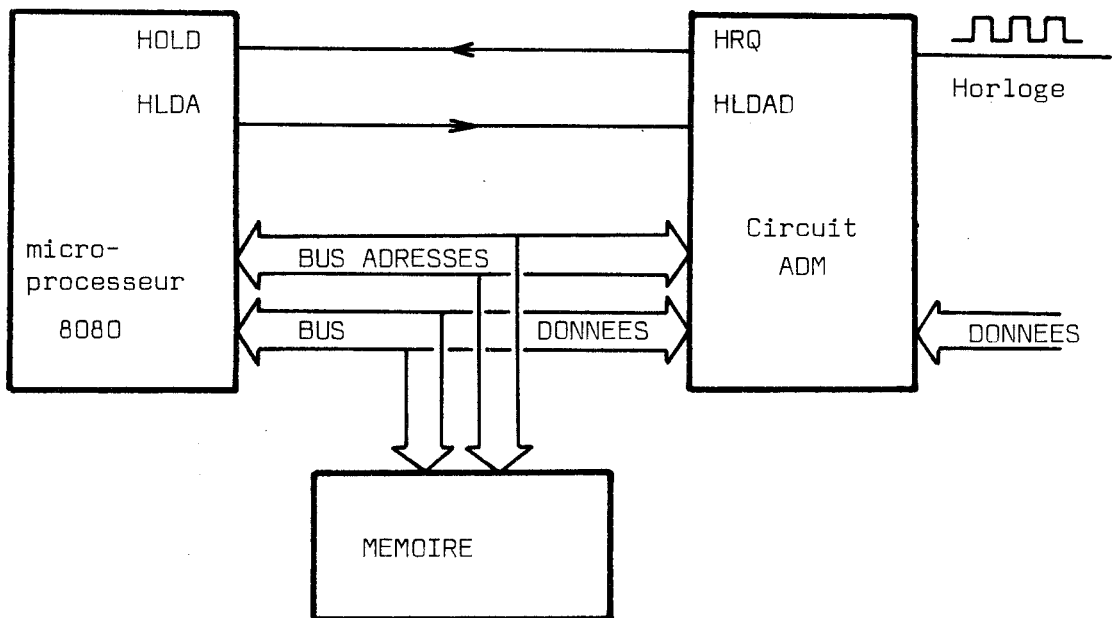


Figure V.9 : Exemple de connexion directe du circuit ADM au 8080.

La commande du HOLD par HRQ permet de "geler" le microprocesseur, c'est-à-dire de le laisser terminer l'instruction en cours, et d'empêcher le déroulement de la suite du programme. Le "gel" du microprocesseur et, par conséquent, la libération des bus sont signalés par l'activation de HLDA.

Cette configuration n'est pas satisfaisante car toutes les 10 μ s, le microprocesseur est gelé pendant 2 μ s, temps requis pour le transfert d'une donnée par l'ADM, d'où les améliorations proposées dans le paragraphe suivant.

V.5.3. Connexion de l'ADM par multiprocessing

V.5.3.1. Principe

La figure V.10 schématise les connexions du circuit ADM au micro-calculateur, tous deux considérés comme processeurs en parallèle.

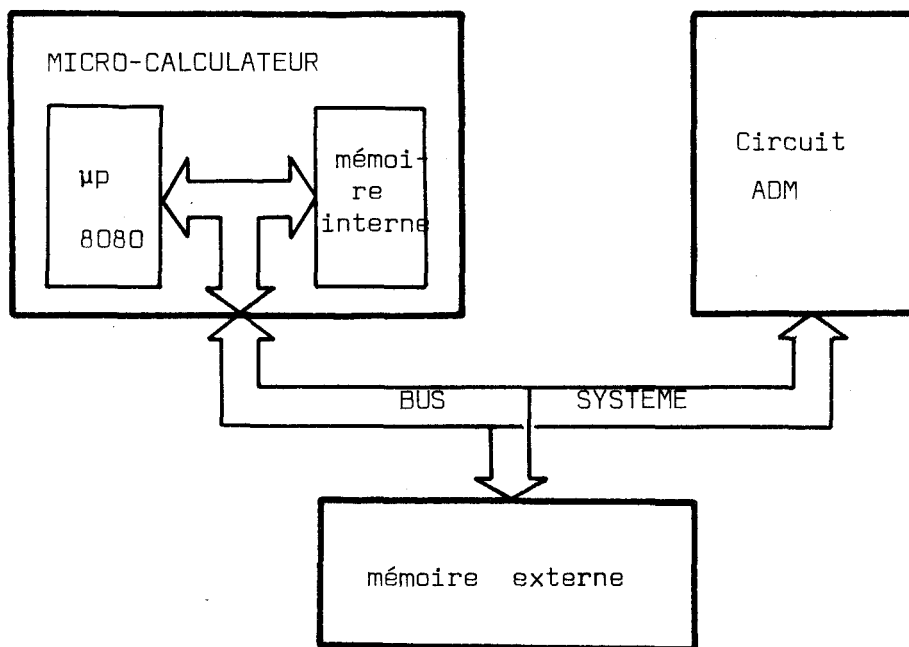


Figure V.10 : Connexion ADM micro-calculateur

La configuration adoptée permet le travail simultané des deux processeurs :

- l'ADM accède uniquement à la mémoire externe constituée de 1024 x k mots (acquisition de k lignes).
- le microprocesseur travaille dans la mémoire interne du micro-calculateur.

Les échanges d'informations sont réalisés de la mémoire externe vers la mémoire interne par l'intermédiaire du microprocesseur.

En implantant les programmes de traitement et les données intermédiaires de travail dans la mémoire interne, le microprocesseur n'est que très peu ralenti par l'acquisition. En effet, le microprocesseur n'est bloqué par l'ADM que lorsque les données $T(i,j)$ des sous-images sont traitées.

V.5.3.2. Occupation du bus système

Il est utile de différencier bus-système et bus du micro-calculateur. Le bus interne au calculateur reste constamment relié à l'unité centrale, alors que les deux bus ne sont connectés que lors d'un adressage d'un mot mémoire extérieur au micro-calculateur. Cette connexion est réalisée après un test d'occupation du bus système. De façon similaire au test d'occupation généré par le circuit ADM, ce test est réalisé par deux signaux :

- BREQ : activé par une demande d'occupation du bus système
- BPRN : réponse de l'extérieur activée quand le bus système n'est pas occupé.

Ce fonctionnement tout à fait symétrique des deux processeurs nous a conduit à adapter une structure également symétrique. Aucune priorité n'est accordée à chacune des unités. Lorsqu'un processeur demande l'occupation du bus, deux cas peuvent se présenter :

- le bus est occupé : le demandeur attend la fin du transfert opéré par l'autre processeur et occupe le bus à son tour.

La figure V.11 montre le caractère symétrique de cette procédure.

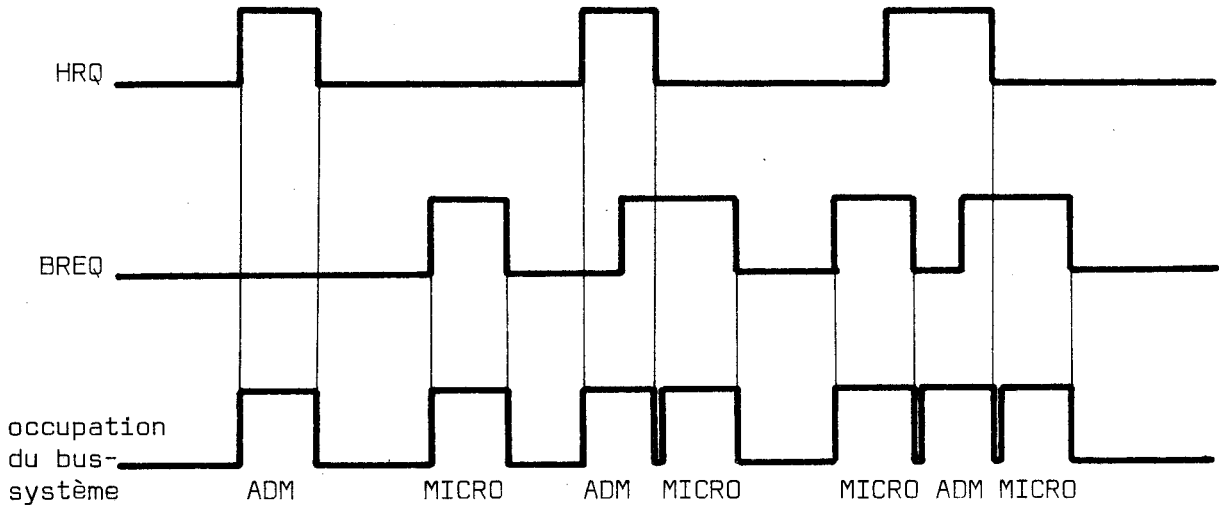


Figure V.11 : Gestion des priorités

Il en découle une réalisation très simple de la gestion d'occupation du bus-système (figure V.12). Un bistable sert de base au contrôleur de priorité.

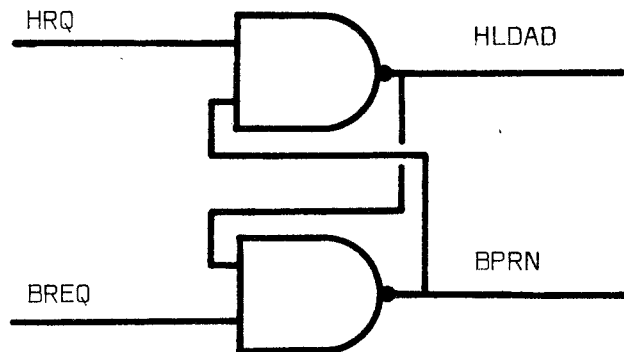


Figure V.12 : Circuit de base du contrôleur de priorité.

Lorsque les deux demandes HRQ et BREQ arrivent simultanément, aucune ambiguïté ne peut survenir à cause de la nature stable de ce circuit.

V.5.3.3. Programmation et initialisation de l'ADM

Cette partie est consacrée à la résolution des problèmes de synchronisation de l'ADM avec le micro-calculateur.

Il est possible de choisir, parmi les nombreuses variantes d'initialisation de l'ADM, un mode de fonctionnement permettant d'initialiser définitivement le circuit d'accès direct mémoire. Ce circuit mémorise en permanence l'adresse mémoire initiale de la zone de transfert et le nombre de transferts. En fin de cycle complet, ces valeurs sont automatiquement rechargées et permettent à l'ADM d'entamer un nouveau cycle sans intervention du microprocesseur.

Pour que cette possibilité soit applicable à notre cas, un cycle complet doit comprendre $1024 \times k$ transferts.

La figure V.13 schématise la procédure employée pour initialiser le système complet.

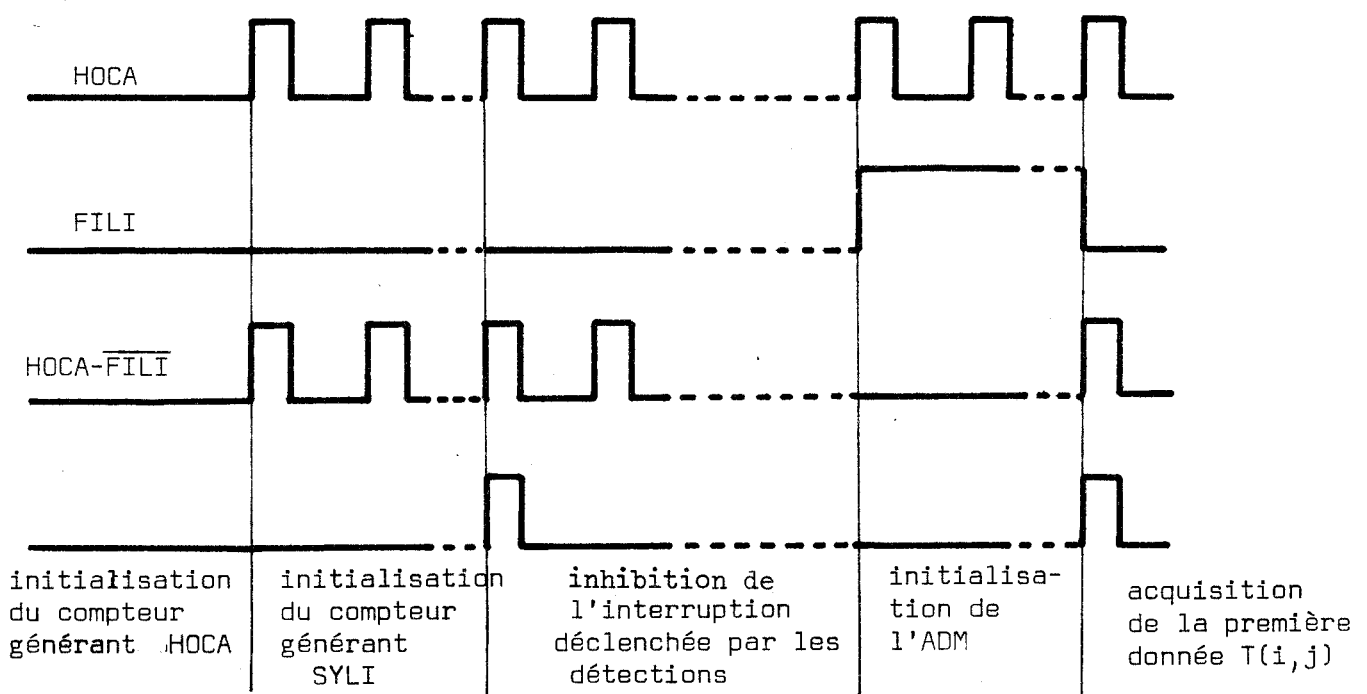


Figure V.13 : Initialisation du système

Dans un premier temps, les compteurs programmables sont initialisés. Comme il n'est pas possible d'initialiser en même temps l'ADM, la procédure générale d'acquisition et de traitement des images n'est pas entamée dès la première ligne.

Le compteur programmable générant le signal SYLI a été initialisé à la valeur $1024 + q$. Le temps compris entre la fin de scrutation de 1024 photodiodes et le début d'analyse de la prochaine ligne est mis à profit pour initialiser l'ADM. Ceci explique le fait que FILI soit le signal d'interruption cyclique (§ V.4.2). De cette façon le signal de la première photodiode est transféré avec certitude dans la première case de la mémoire image.

Pour que l'ADM ne transfère que 1024 données par ligne au lieu de $1024 + q$, le signal de synchronisation de l'ADM est HOCA . FILI.

V.6. Conclusion

Nous venons de présenter, dans ce chapitre, la liaison entre une caméra et un micro-calculateur, ensemble qui constitue un capteur d'images défilant à grande vitesse.

La constitution et l'acquisition de l'image fugitive, formée de k lignes d'informations constamment renouvelées sont effectuées entièrement par voie matérielle; ceci permet un traitement des sous-images transférées de la mémoire externe vers la mémoire interne, sans perte de temps. De plus, une gestion des programmes par interruptions logicielles réduit la taille mémoire réservée à ces sous-images. Cette gestion logicielle fait l'objet du chapitre suivant.

CHAPITRE VI

- IMPLICATIONS LOGICIELLES.

Ce chapitre aborde le problème de la transposition logicielle des méthodes d'acquisition et de traitement d'images. Pour réduire la place mémoire destinée aux images, sous-images et résultats intermédiaires, nous avons été amené à décomposer la procédure en tâches ayant des niveaux de priorité différents. La gestion de ces différentes étapes est effectuée par des interruptions logicielles.

Les parties VI.2, VI.3, VI.4 sont consacrées à une étude plus détaillée de chaque tâche. En particulier, plusieurs méthodes de minimisation de taille de fichiers sont étudiées.

VI.1 Décomposition de la procédure totale en tâches de niveau de priorité différent

VI.1.1. Implications du traitement en temps réel

Les procédures d'acquisition et de traitement d'images, développées dans la première partie de ce mémoire, s'appliquent à des images défilant très rapidement et de façon ininterrompue. Certes, la fréquence d'apparition des formes est faible : environ 300 défauts pour 30 minutes d'analyse. Mais ces formes apparaissent de façon tout à fait aléatoire. Le prototype doit donc être capable de supporter, sans saturation, une densité élevée de défauts pendant un temps très court.

Etant données la relative lenteur du matériel informatique et la masse importante d'informations, il est bien souvent impossible de traiter toutes les données au moment de leur acquisition. Les traitements sont alors effectués en temps légèrement différé, sur les données préalablement mémorisées.

La procédure temps réel qui va être exposée a pour rôle de coordonner les opérations logicielles afin de limiter cette mémoire tampon à une taille acceptable.

VI.1.2. Description des tâches

La minimisation de la mémoire tampon s'associe à la notion de tâche prioritaire. Citons un exemple : la paramétrisation et la reconnaissance des impuretés sont des traitements longs qui s'effectuent sur peu de données (le contour des formes). Par contre, la validation des détections se réalise très rapidement sur un volume de données important. Lorsque des détections apparaissent lors de la paramétrisation d'une forme, cette tâche est temporairement abandonnée, pour éviter que les données de l'image associées aux détections ne s'accumulent en mémoire.

La figure VI.1 représente les différentes phases du traitement qui, à partir de l'image initiale, conduisent à la détection et à la reconnaissance des défauts. Appelons $Tr(i)$ chaque étape ayant pour rôle de réduire le nombre d'informations d'une image $I(i)$ à une image $I(i+1)$. A chacune de ces transformations $Tr(i)$, nous avons associé un taux de réduction des informations : rapport du nombre initial de données au nombre de données après application de $Tr(i)$. Les vitesses d'exécution des différentes transformations sont également portées sur la figure VI.1. Les taux de réduction et vitesses de traitement ne sont données qu'à titre d'exemple; en effet, ces paramètres fluctuent en fonction de l'homogénéité du voile de laine et de la nature des défauts.

Etant donné que chaque étape contribue à une réduction d'informations et que les vitesses d'exécution diminuent d'une transformation $Tr(i)$ à la suivante $Tr(i+1)$, nous en déduisons le degré de priorité de chacune des tâches : une transformation $Tr(i)$ est prioritaire sur les étapes $Tr(i+k)$, k positif.

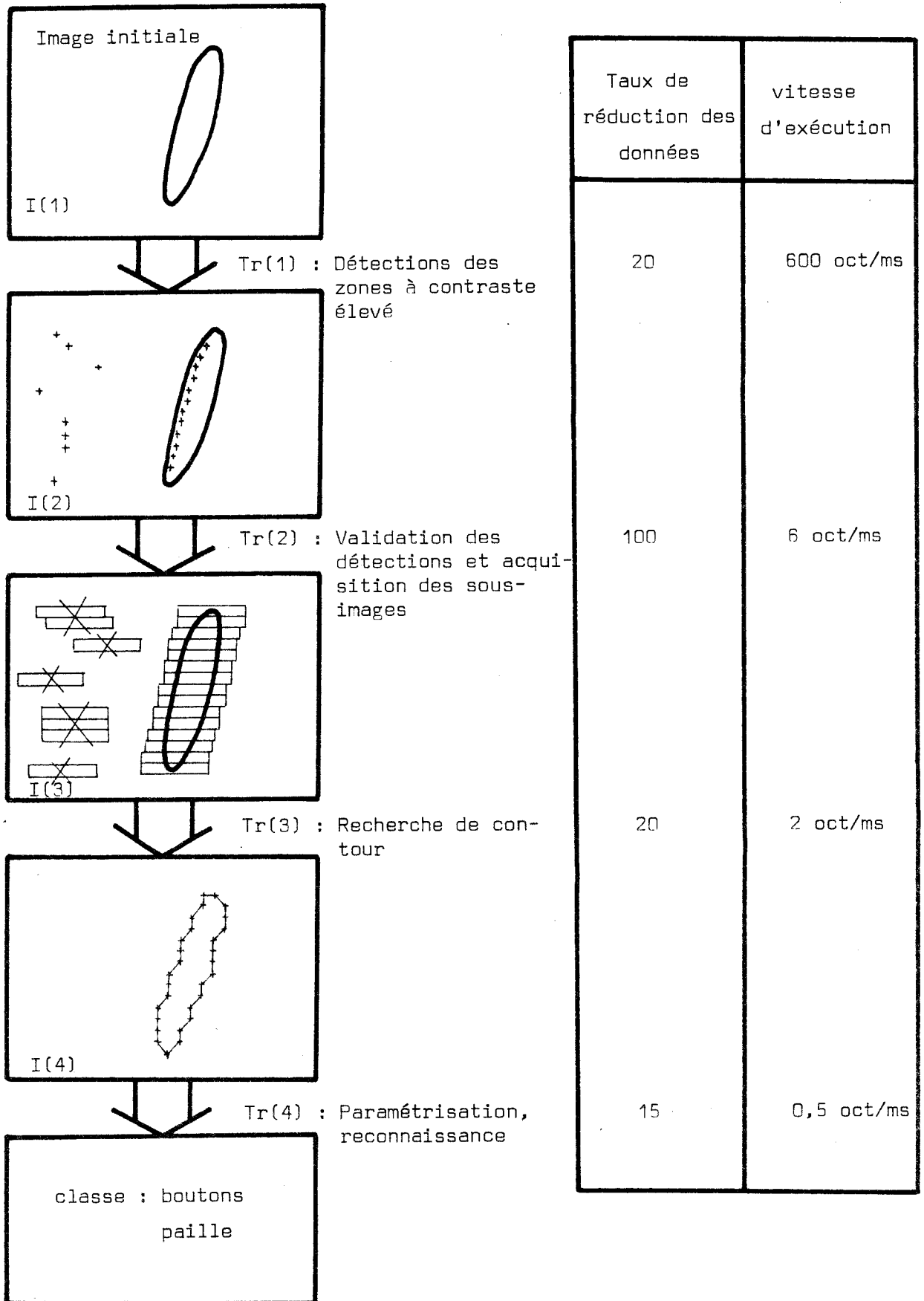


Figure VI,1 : Décomposition de la procédure totale

VI.1.3. Imbrications des tâches

Il n'est pas nécessaire qu'une sous-image de défaut soit complètement acquise pour débiter la recherche de contour. En effet, la détection de contours s'effectue ligne par ligne (chapitre III). Pour un même défaut, les tâches d'acquisition et de recherche de contour peuvent donc se réaliser parallèlement. Cette imbrication des tâches est particulièrement apparente dans les zones d'image à forte densité d'impuretés.

Soient trois défauts dont les références indiquées sur la figure VI.2 représentent l'ordre dans lequel ils sont acquis. Le défaut n°1 étant acquis, supposons qu'il soit possible d'effectuer la détection de contour d'une partie de la paramétrisation de défaut avant que la forme n°2 apparaisse. La tâche de paramétrisation est abandonnée temporairement et remplacée par l'acquisition et la détection de contour de la deuxième impureté.

Remarquons sur la figure VI.2. que les défauts n°2 et 3 apparaissent partiellement sur les mêmes lignes de l'image. L'acquisition simultanée des deux sous-images oblige à abandonner pratiquement la tâche de détection de contours du défaut n°2. Lorsque l'acquisition des deux sous-images est terminée; les tâches se déroulent dans cet ordre :

- fin de recherche de contour de 2 et 3.
- fin de paramétrisation de 1.
- paramétrisation de 2, puis de 3.

Toutes ces opérations sont entrecoupées par des tâches de validation visant à éliminer les fausses détections. Pour ne pas encombrer la figure VI.2, seulement deux fausses détections sont représentées.

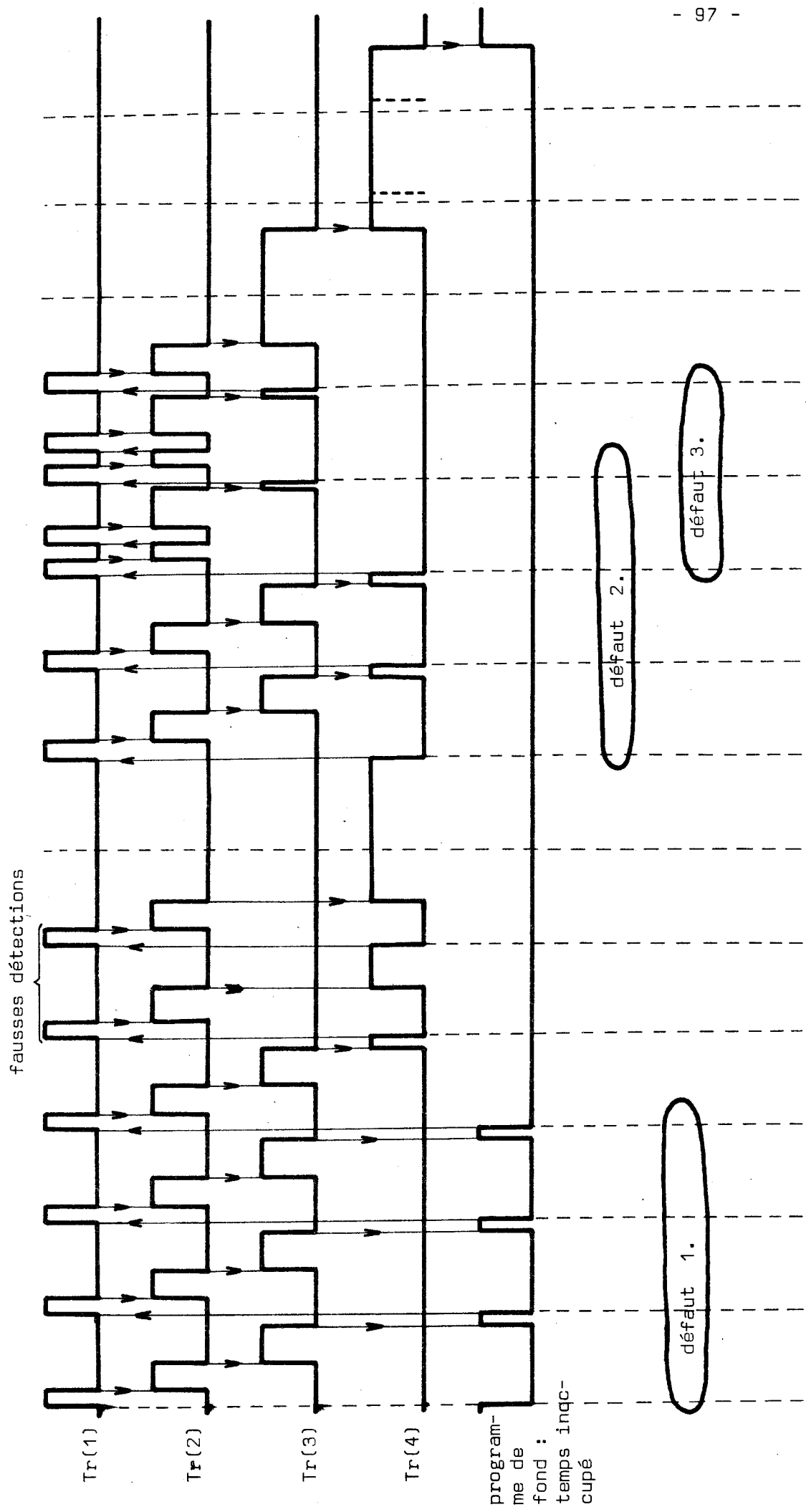


Figure VI.2 : Exemple d'acquisition et de traitement de trois défauts.

VI.1.4. Mémorisation des données intermédiaires.

Le résultat de chaque transformation d'image $Tr(i)$ doit être mémorisé temporairement pour être traité a priori par la transformation $Tr(i+1)$. Cette mémoire est une file d'attente, car les dernières informations mémorisées après transformations $Tr(i)$ sont traitées par $Tr(i+1)$ en dernier lieu.

La figure VI.1. représente les différentes files d'attente utilisées :

- M_1 mémorise les données $T(i,j)$ des zones à contraste élevé de l'image.
- M_2 mémorise les sous-images des formes
- M_3 mémorise les contours des formes.

A chacune des mémoires M_i , sont associés deux pointeurs : Pt_s^i et Pt_b^i . Le pointeur de sommet Pt_s^i indique l'endroit de la file d'attente M_i , où sont rangés les prochains résultats de l'étape $Tr(i)$. Le pointeur de base de la file d'attente Pt_b^i , utilisée par la transformation $Tr(i+1)$ indique l'adresse des données de M_i à traiter en priorité.

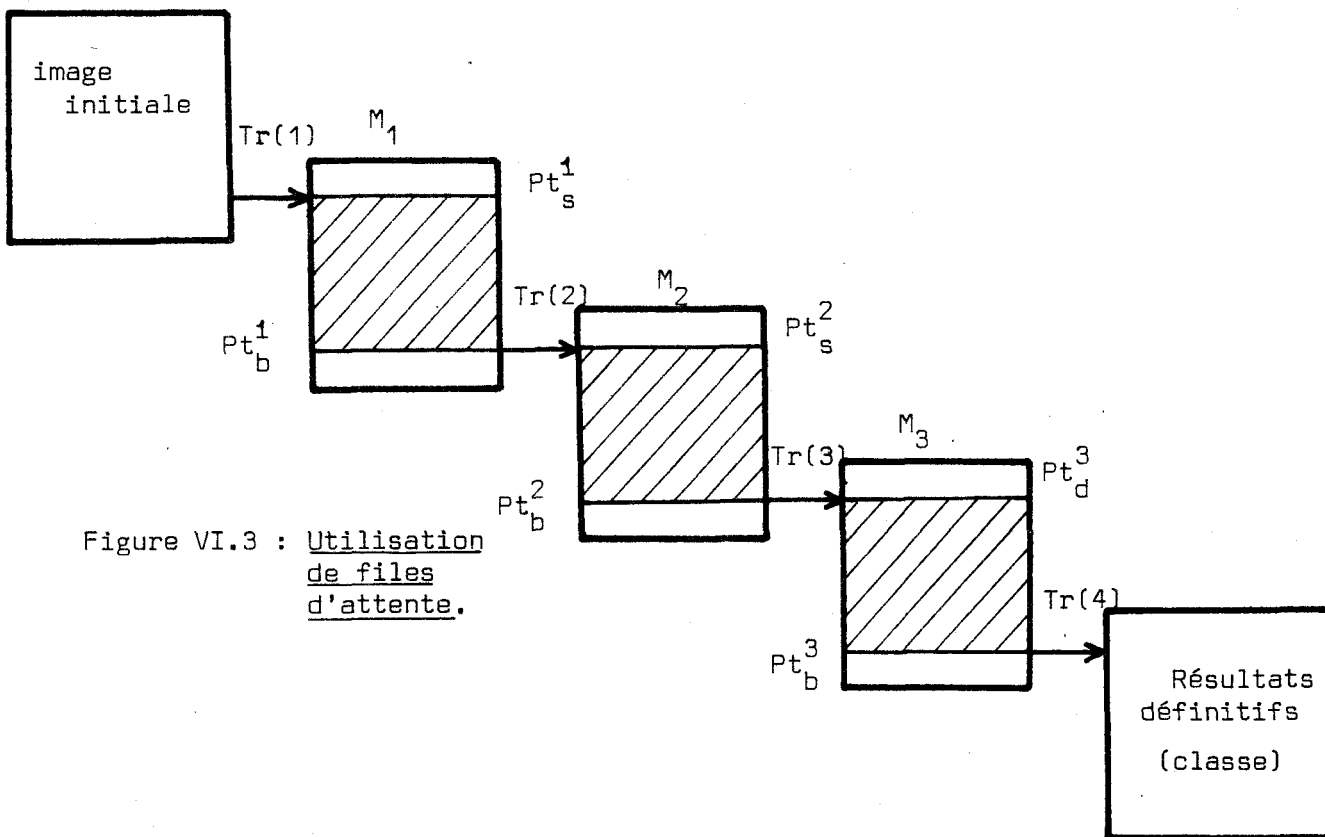


Figure VI.3 : Utilisation de files d'attente.

La gestion du traitement total a pour rôle de tester en permanence l'état de chacune des piles, commençant, en priorité, par M_1 . Celle-ci est vidée en effectuant la transformation $Tr(2)$, puis M_2 et M_3 subissent le même traitement. Le cycle complet est représenté sur la figure VI.4.



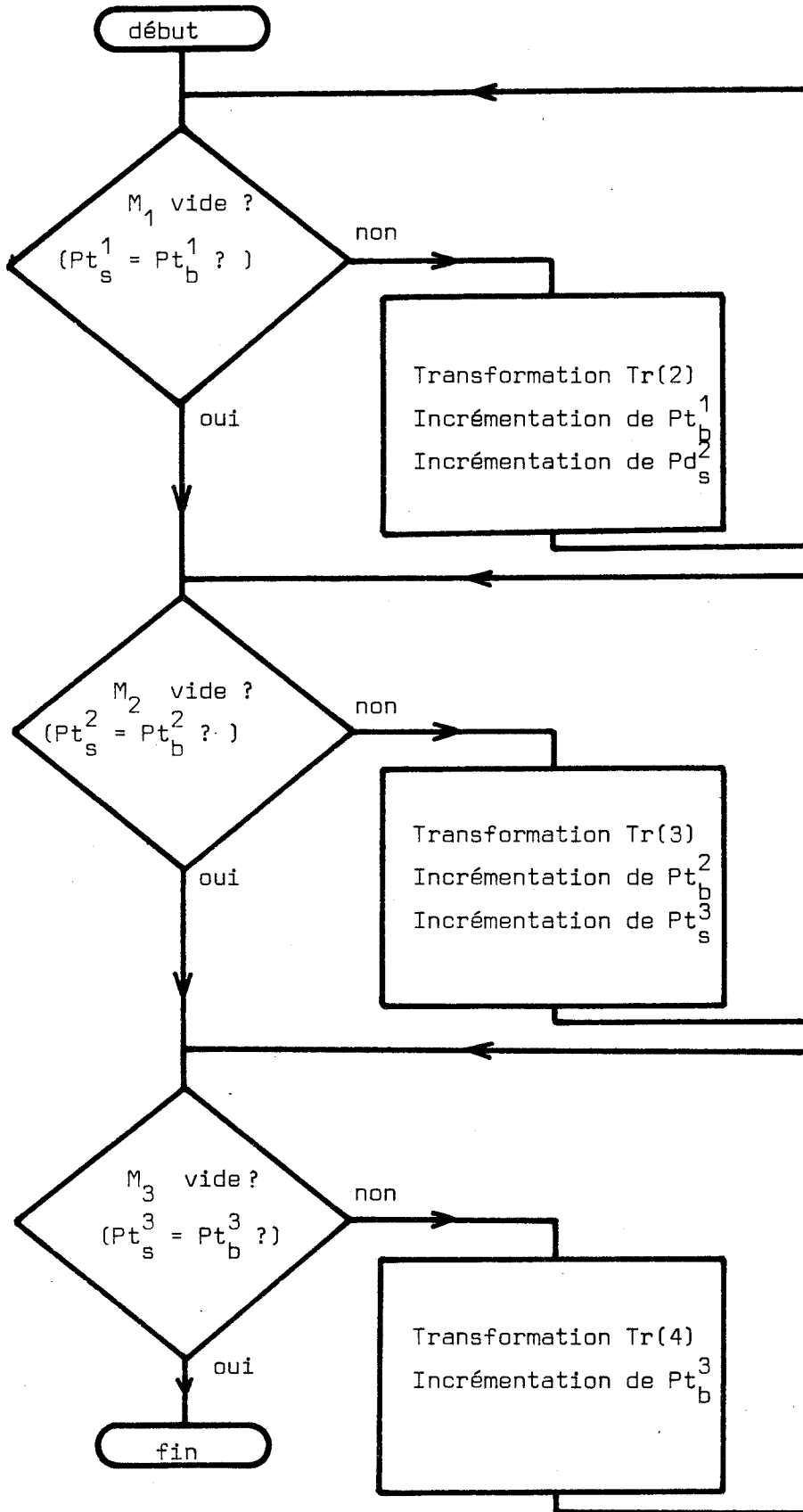


Figure VI.4 : Gestion des files d'attente

VI.1.5. Interruptions logicielles

La détection des zones de l'image à contraste élevé, entraînant le chargement de la mémoire M_1 , survient de façon aléatoire. Il est donc nécessaire de contrôler périodiquement l'état de M_1 , pour en assurer la remise à zéro en employant la transformation $Tr(2)$. Ce contrôle cyclique est exécuté grâce à une interruption matérielle commandée par une horloge temps réel.

Un programme d'interruption est un sous-programme dont l'appel, au lieu d'être logiciel, est réalisé par voie matérielle. Cela suppose un retour à un programme de fond lorsque le contrôle de l'état des mémoires M_i est terminé. Le programme de fond se résoud, tout simplement à une boucle d'attente. En effet, la procédure totale est exécutée par des interruptions :

- $Tr(1)$ est appelée grâce à une interruption de type aléatoire (détection de zone à contraste élevé)

- $Tr(2)$, $Tr(3)$, $Tr(4)$ sont appelés par l'intermédiaire d'une interruption cyclique.

L'organigramme schématisé par la figure VI.4 n'est pas applicable directement. En effet, les temps d'exécution d'une transformation $Tr(i)$ peuvent être plus longs qu'un cycle de l'horloge temps réel. Il faut donc laisser la possibilité au programme d'interruption temps réel d'être interrompu par lui-même. La modification apportée à la gestion des interruptions logicielles est représentée sur la figure VI.5.

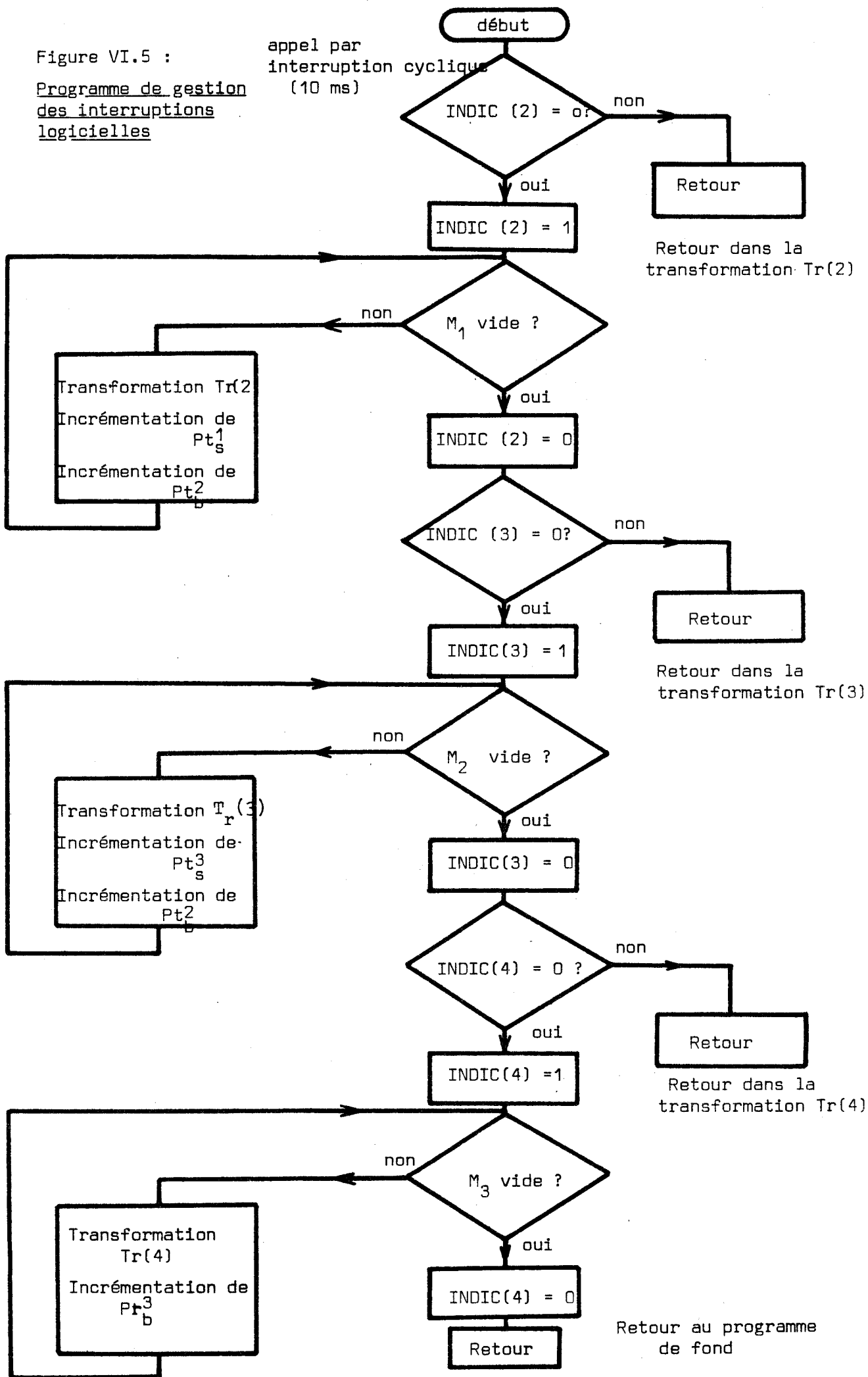
A chaque transformation $Tr(i)$ est associé un indicateur INDIC (i) dont le rôle est de signaler si $Tr(i)$ est en cours d'exécution. Si, après un arrêt de $Tr(i)$ par l'interruption temps réel, aucune tâche de priorité supérieure ($Tr(i-1)$, $Tr(i-2)$...) ne doit être effectuée, le test de l'indicateur INDIC (i) permet un retour immédiat dans le programme $Tr(i)$.

Le temps de cycle de l'horloge temps réel est choisi égal au temps d'analyse de la ligne de photodiodes (10 ms). Le signal d'interruption est FILI : indicateur de fin d'analyse des 1024 photodiodes (§ V.4.2). Le choix de ce signal a permis, outre la gestion des interruptions logicielles :

- l'incréméntation de l'indice de ligne i .
- le contrôle de l'acquisition des défauts : nous verrons qu'après avoir vidé la file d'attente M_1 , il est nécessaire de contrôler l'acquisition des sous-images (§VI.2).

Figure VI.5 :
Programme de gestion
des interruptions
logicielles

appel par
interruption cyclique
(10 ms)



VI.2. Etude de la transformation Tr(1)

VI.2.1. Description de la file d'attente M_1

La transformation Tr(1) a pour rôle de mémoriser, pour chaque détection survenue au point d'image (i, j_d^i) , les pixels $T(i, j)$; $j \in [j_d^i - \Delta_1, j_d^i + \Delta_2]$. Le transfert s'effectue de la mémoire adressée par le système à accès direct mémoire (image fugitive) vers la file d'attente M_1 . Les données de M_1 sont traitées par Tr(2) : elles sont soit rejetées, si la détection n'appartient pas à un défaut, soit incluses à un défaut en cours d'acquisition, ou constituent les premières données d'une nouvelle sous-image. Dans ce dernier cas, les informations mémorisées dans la mémoire M_1 ne suffisent pas, car le cadrage du défaut impose la reconnaissance des pixels proches de j_d^i sur les lignes $i-1, i-2$, c'est-à-dire $T(i-k, j)$, $j \in [j_d^i - \Delta_1, j_d^i + \Delta_2]$, $k \in [0, 2]$.

Une première solution consiste à mémoriser systématiquement ces données dans la file d'attente M_1 . Les performances de vitesse d'analyse seraient néanmoins très limitées par cette solution. En effet, le nombre de fausses détections est élevé par rapport au nombre de détections validées : le taux de fausses détections est de l'ordre de 99%. Donc la presque totalité des données seraient transférées inutilement de la zone image fugitive vers la mémoire M_1 .

Pour éviter cet inconvénient, la mémoire M_1 est réservée uniquement au stockage des positions (i, j_d^i) des détections. La transformation Tr(2) s'effectue directement à partir des données de l'image fugitive.

Cette solution apporte un gain appréciable de vitesse, mais elle nécessite la mémorisation permanente de p lignes de l'image. En effet, il se peut que, lors de l'acquisition de la ligne d'indice i , la validation des détections s'effectue sur les données d'une ligne d'indice $i-k$. Ce retard peut être dû à une forte densité de détections sur les quelques lignes $i-p, \dots, i$. La figure VI.6 illustre le rôle de mémoire tampon constituée par p lignes mémorisées en permanence.

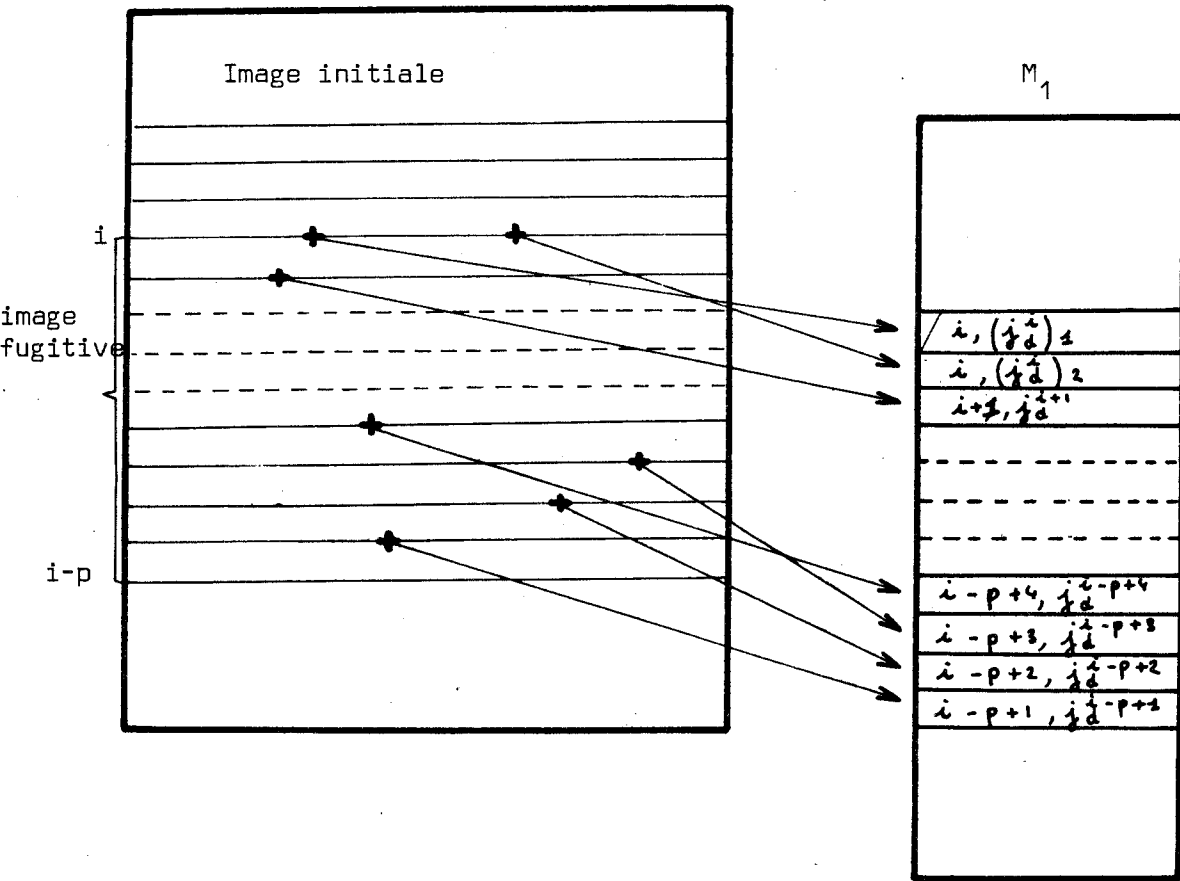


Figure VI.6 : Image fugitive.

La mémorisation des lignes de l'image d'indice $i-k$, $k \in [0, p]$ permet la validation de détections apparues uniquement sur les lignes d'indice $i-k'$, $k' \in [0, p-2]$. En effet, lorsqu'une détection est validée sur la ligne $i-p+2$, le cadrage d'un nouveau défaut nécessite la mémorisation des deux lignes d'indice $i-p+1$, $i-p$.

Dans notre application, une image fugitive constituée de 8 lignes ($p=7$) permet de faire face aux plus fortes densités de détections apparues lors des essais.

VI.2.2. Gestion de la file d'attente M_1

Le mode d'acquisition des coordonnées des détections (i, j_d^i) conditionne la constitution de la mémoire M_1 . Le système d'acquisition doit permettre la distinction entre deux défauts voisins sur une même ligne. Pour cela, le programme d'interruption activé par l'impulsion logique de détection (§ II.3) doit être le plus court possible. La figure VI.7 montre que l'association d'un programme d'acquisition très court et d'un traitement de données situées de part et d'autre des positions j_d^i , permet de ne pas avoir de "trou" entre deux détections rapprochées.

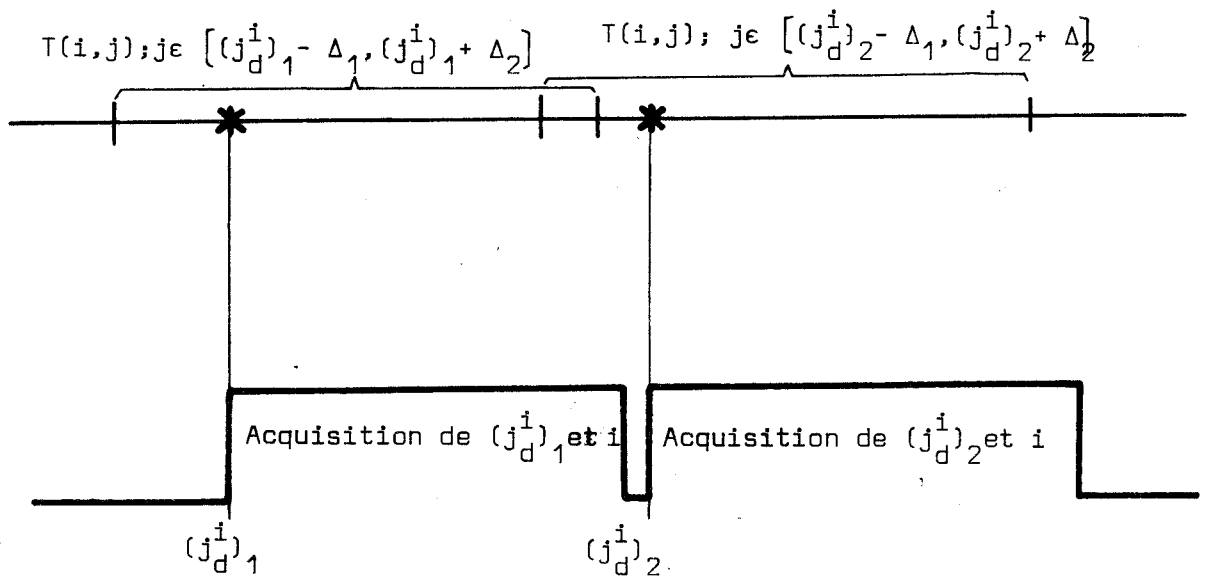


Figure VI.7 : Acquisition de deux segments de forme très proches.

Pour réduire le temps d'exécution du programme d'interruption, la gestion de la mémoire M_1 ne s'effectue pas lors de l'acquisition de la position des détections, mais lors de la transformation $Tr(2)$. La gestion de M_1 a pour rôle d'éviter que les pointeurs Pt_s^1 et Pt_b^1 ne débordent de la mémoire M_1 . Lorsque la file d'attente est vide, c'est-à-dire $Pt_s^1 = Pt_b^1$, ces deux pointeurs sont réinitialisés en début de mémoire M_1 . La figure VI.8 montre l'évolution des pointeurs, au cours du temps.

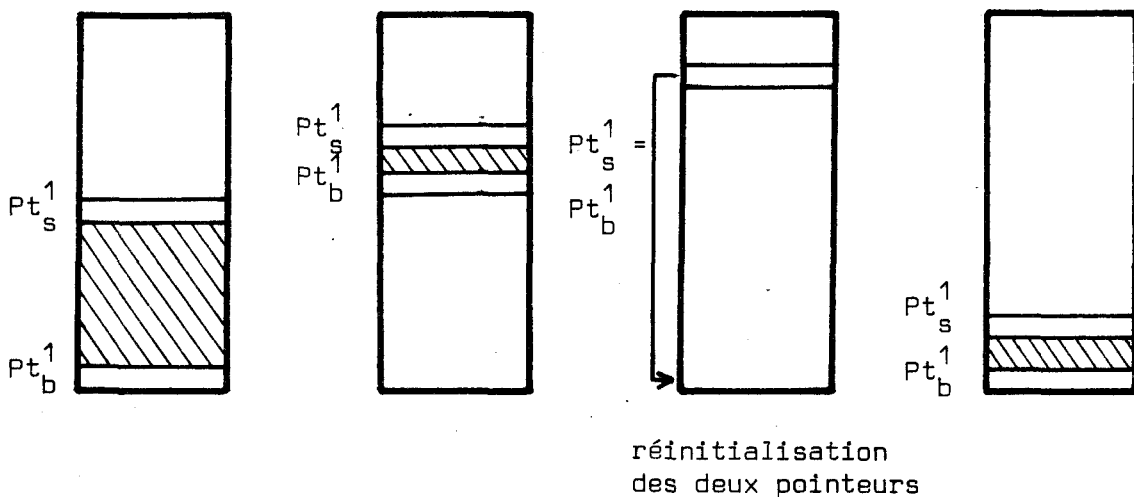


Figure VI.8 : Evolution des pointeurs de M_1 .

VI.3. Etude de la transformation $Tr(2)$

VI.3.1. Organisation de la file d'attente M_2

Le but de la transformation $Tr(2)$ est de valider les détections $T(i,j)$, $j \in [j_d^i - \Delta_1, j_d^i + \Delta_2]$ contenues dans l'image fugitive constituée de 8 lignes. Le calcul de l'adresse de rangement de ces informations s'effectue à partir de (i, j_d^i) rangées dans la mémoire M_1 .

Soit $TAB(k,l)$, $k \in [0,7]$, $l \in [1, 1024]$ une représentation matricielle de la mémoire de l'image fugitive. Une relation très simple associe $T(i,j)$ et $TAB(k,l)$

$$T(i,j) = TAB(i(\text{modulo } 8),j).$$

Cette relation exprime le fait que l'image est mémorisée par blocs glissants de 8 lignes.

Si une détection est validée ou appartient à une forme dont la sous-image est en cours d'acquisition, les pixels $T(i,j)$ correspondant à cette détection seront alors extraits de l'image fugitive et stockés dans la file d'attente M_2 . Cette mémoire est organisée en secteurs adressés par Pt_s^2 lors de l'acquisition d'une partie de sous-image, par Pt_b^2 lors du traitement d'un segment de forme par $Tr(3)$.

Pour diminuer la place mémoire requise par cette file d'attente, on profite de la place libre, située entre le début de M_2 et l'adresse pointée par Pt_b^2 . Lorsque la mémoire M_2 est remplie, c'est-à-dire que Pt_s^2 dépasse la taille maximale de la mémoire, ce pointeur est réinitialisé en début de la file d'attente. La figure VI.9 montre l'évolution des pointeurs, au cours du temps.

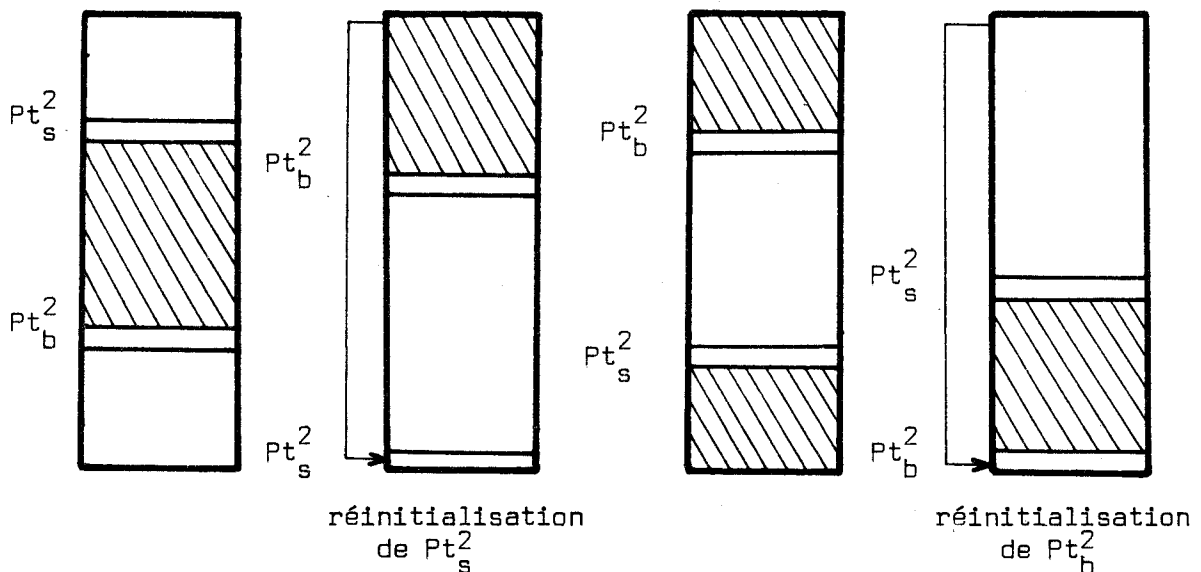


Figure VI.9: Evolution des pointeurs de M_2

Cette rotation des pointeurs Pt_s^2 et Pt_b^2 ne peut s'effectuer sans précaution. Il faut s'assurer que la file d'attente n'est pas saturée, c'est-à-dire que Pt_s^2 ne dépasse pas Pt_b^2 . Les exemples de la figure VI.9 montrent qu'une simple comparaison entre les deux pointeurs ne suffit pas. Pour réaliser ce test, chaque secteur comprend un indicateur d'occupation (= 0 : secteur libre, $\neq 0$: secteur occupé). L'incrémentatation du pointeur Pt_s^2 s'effectue après avoir testé l'occupation du secteur. Si celui-ci est occupé, M_1 est saturée.

VI.3.2. Organisation des secteurs de la pile d'attente M_2

Chaque secteur de M_2 contient les informations suivantes :

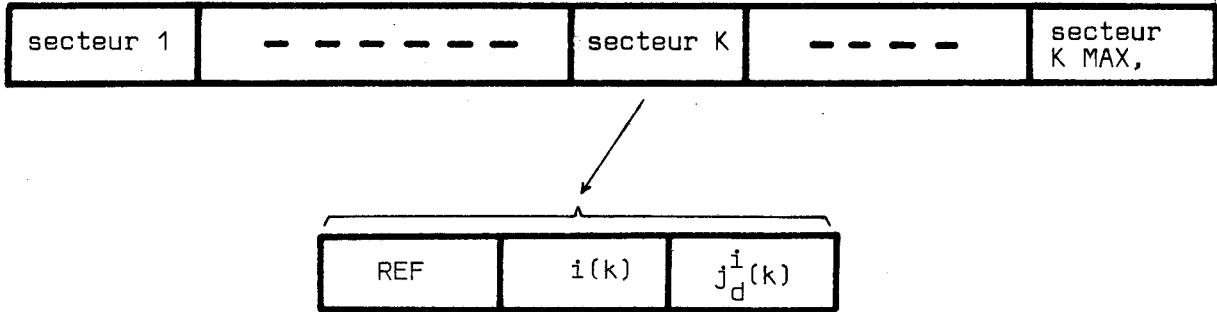
REF	j_d^i	i	$T(i,j)$, $j \in \{ j_d^i - \Delta_1 , j_d^i + \Delta_2 \}$
-----	---------	---	--

L'indicateur REF a deux fonctions :

- si REF = 0, le secteur est libre
- si REF \neq 0, le secteur est occupé par les données d'une sous-image de référence REF. En effet, il est nécessaire de référencer les segments de forme des sous-images. Si plusieurs impuretés apparaissent sur une même ligne, les segments des sous-images, étant acquis séquentiellement, sont imbriqués les uns dans les autres. Chaque forme a donc une référence REF qui est un numéro d'ordre incrémenté à chaque acquisition d'une nouvelle sous-image.

VI.3.3. Test d'appartenance d'un segment de forme à une sous-image en cours d'acquisition

Il est nécessaire de tester l'appartenance d'une détection à chacune des formes dont les sous-images sont en cours d'acquisition. Dans ce but, un fichier contient les principales informations des formes partiellement acquises.



k MAX : nombre total de formes en cours d'acquisition

REF k : référence de la k^{ième} forme

(i(k), j_dⁱ(k)) : coordonnées de la dernière détection de la k^{ième} forme

Une détection de coordonnées (i, j_dⁱ) est incluse à la sous-image de la forme k si :

$$i = i(k) + 1 ; j_d^i \in [j_d^i(k) - \Delta a , j_d^i(k) + \Delta a]$$

Un secteur k du fichier est annulé lorsque l'acquisition de la forme correspondante est terminée. L'exemple de la figure VI.10. montre que l'ordre d'apparition des formes , n'est pas forcément l'ordre dans lequel les acquisitions de sous-images se terminent . Cette figure permet également de visualiser la restructuration du fichier dans un tel cas.

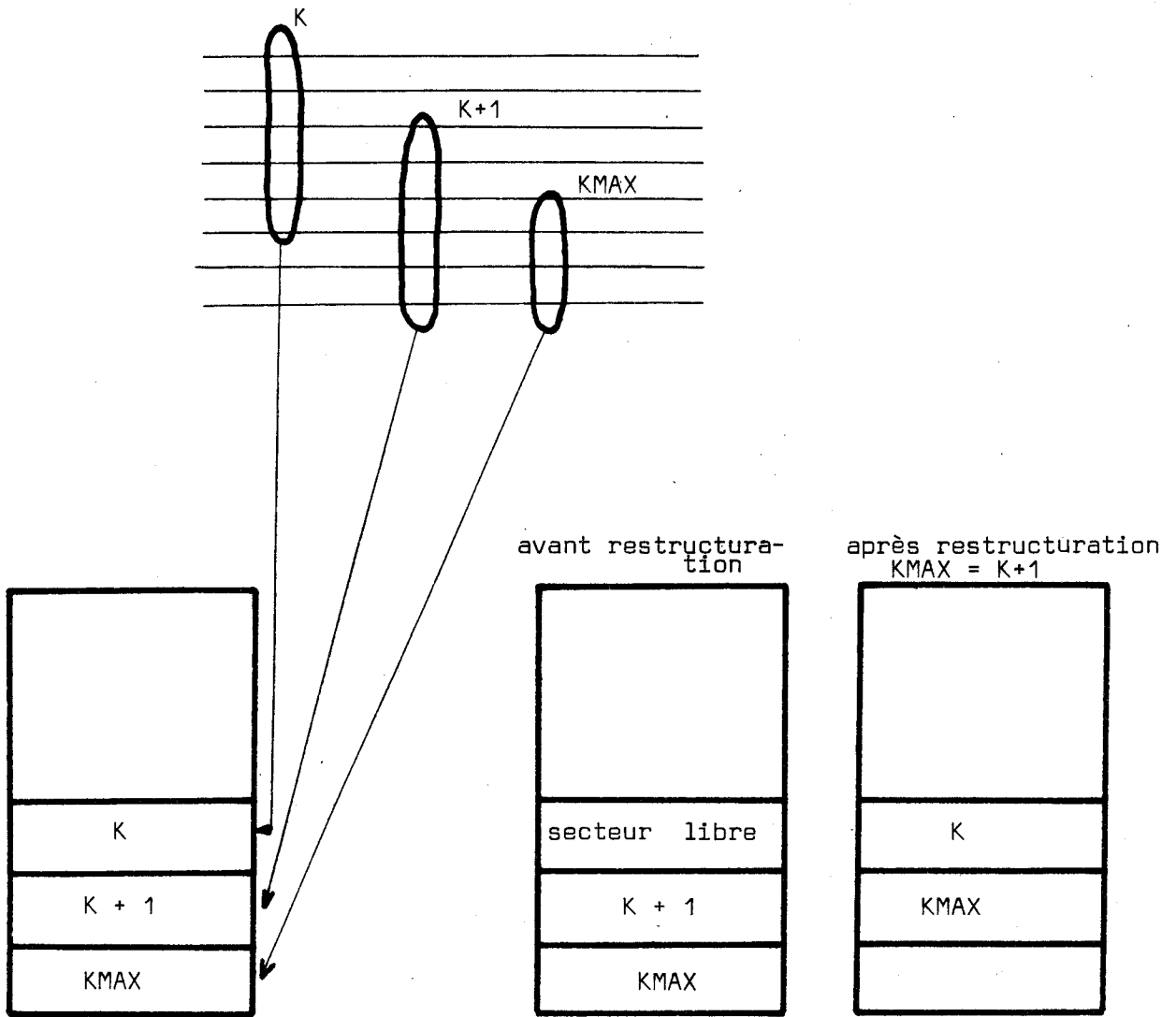


Figure VI.10 : Restructuration du fichier

VI.3.4. Organigramme général de la transformation Tr(2)

La figure VI.11. représente l'organigramme de la transformation Tr(2) comportant trois parties principales :

- validation des détections et initialisation des sous-images (A)
- tests d'appartenance et poursuite de l'acquisition des sous-images (B)
- contrôle de fin d'acquisition des sous-images (C)

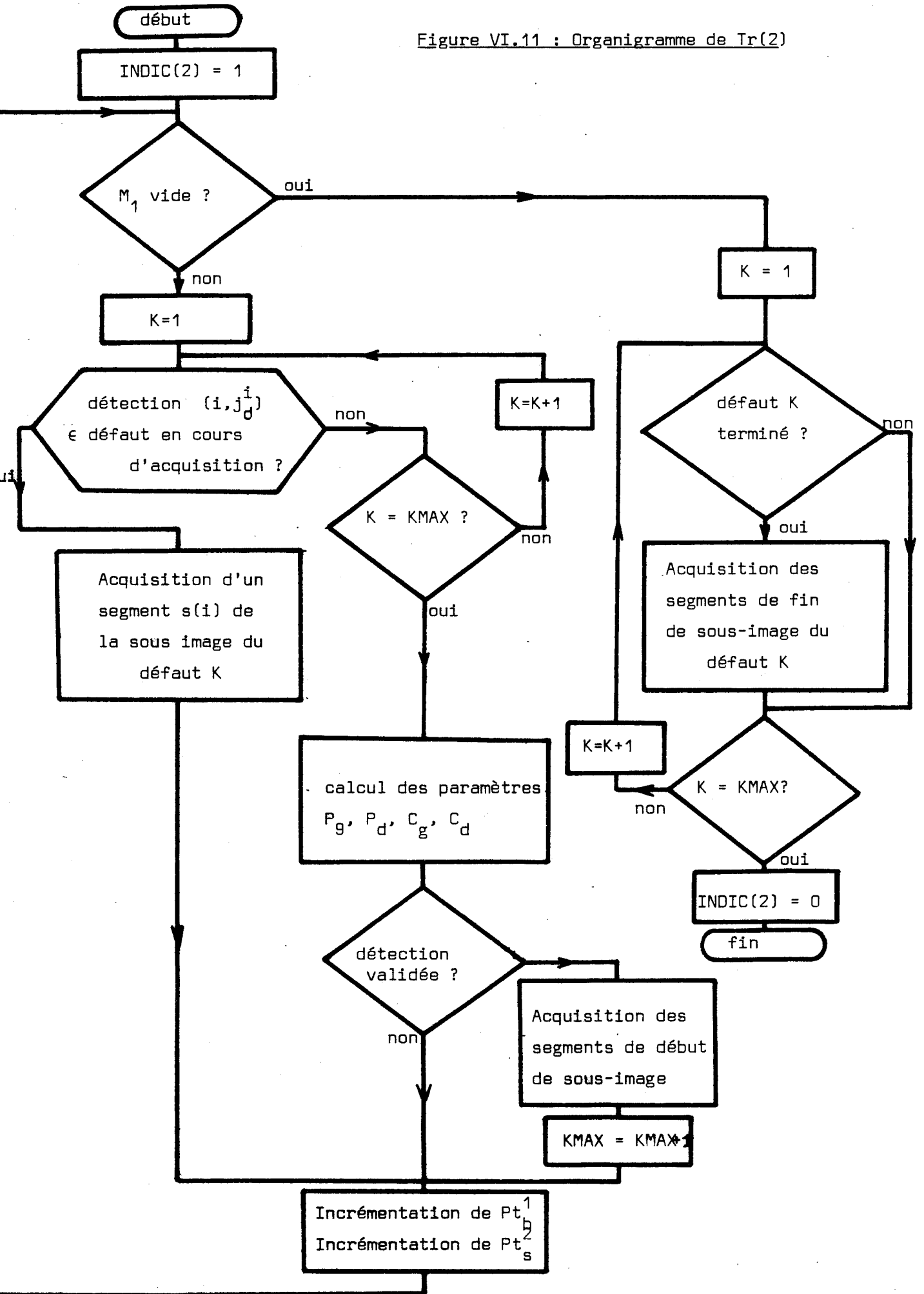
Cette dernière partie a pour rôle de tester si chacune des formes en cours d'acquisition est terminée . L'acquisition d'une forme est achevée lorsque la relation $i - i(k) > 2$ est vérifiée.

i : indice de ligne de l'image qui est en cours d'analyse.

$i(k)$: indice de ligne de la dernière détection de la $k^{\text{ième}}$ forme.

Ce test ne peut être effectué que lorsque la mémoire M_1 est vide. En effet, M_1 peut encore contenir des segments de forme du $k^{\text{ième}}$ défaut, auquel cas l'acquisition de la sous-image de cette forme n'est pas encore achevée.

Figure VI.11 : Organigramme de Tr(2)



VI.4. Etude des transformations Tr(3), Tr(4).

VI.4.1. Organisation de la file d'attente M₃

Le résultat de la transformation Tr(2) est constitué par le contour des formes. Nous avons vu dans le chapitre III que le résultat se divise en deux parties : une chaîne de contour gauche et une chaîne de contour droit. Le nombre d'éléments de ces chaînes varie dans des proportions importantes selon la forme analysée : De façon à minimiser la taille mémoire requise par M₃, la structure de fichier de type séquentiel est préférable au fichier direct. En effet, chaque secteur d'un fichier direct doit avoir une taille suffisante pour contenir toutes les informations du plus grand défaut que l'on peut rencontrer dans un peigné.

La mémoire M₃ est donc constituée de secteurs de petite taille : les derniers octets de chaque secteur indiquent l'adresse du prochain secteur, permettant ainsi de retrouver toutes les informations d'un contour par chaînage. La figure VI.1.2. montre un exemple d'utilisation de ce fichier séquentiel. La connaissance des adresses de début et de fin suffisent à la reconstitution d'une chaîne de contour droit ou gauche.

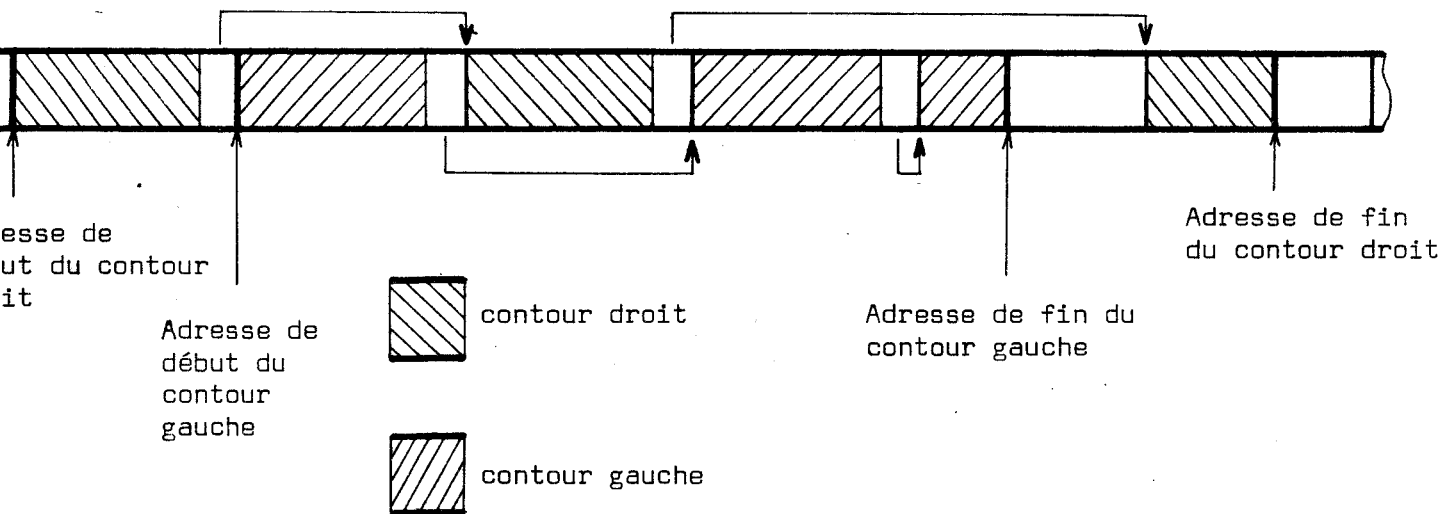
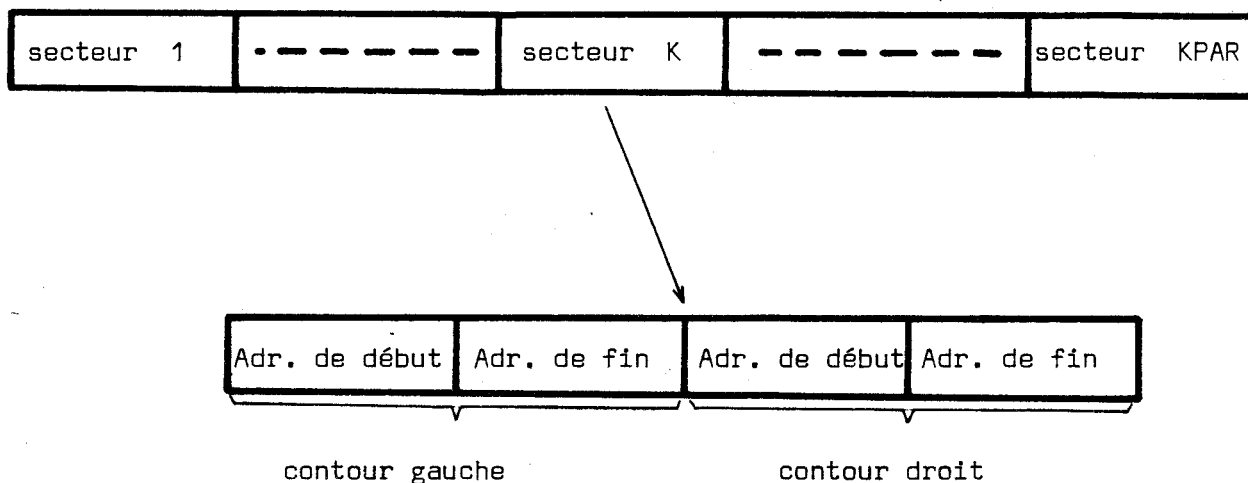


Figure VI.12 : Utilisation d'un fichier séquentiel

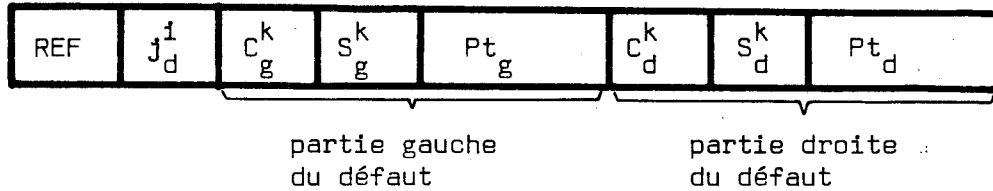
De façon à minimiser la place mémoire de M_3 , nous avons utilisé la même structure que pour M_2 : les pointeurs Pt_s^3 , Pt_b^3 sont réinitialisés en début de M_3 , lorsqu'ils atteignent le haut de la file d'attente. Chaque secteur de M_3 comprend un indicateur d'occupation (0 : le secteur est libre, 1 : le secteur est occupé).

A la file d'attente M_3 est associé un fichier contenant les informations de chaque forme, nécessaires à la transformation $Tr(4)$. Le nombre de secteurs de ce fichier $KPAR$, correspond au nombre de formes dont le contour est déjà extrait, mais qui n'ont pas subi la transformation $Tr(4)$ (paramétrisation). Chaque secteur de ce fichier contient les adresses de début et de fin des chaînes de contour gauche et droit.



VI.4.2. Fichier intermédiaire pour la recherche de contour

Au cours de la recherche des frontières d'un segment de forme $s(k+1)$, il est nécessaire de connaître certaines informations concernant le segment de forme $s(k)$. A cette fin, un fichier est constitué comprenant autant de secteurs que de formes dont la recherche de contour est en cours. Chaque secteur contient les informations suivantes :



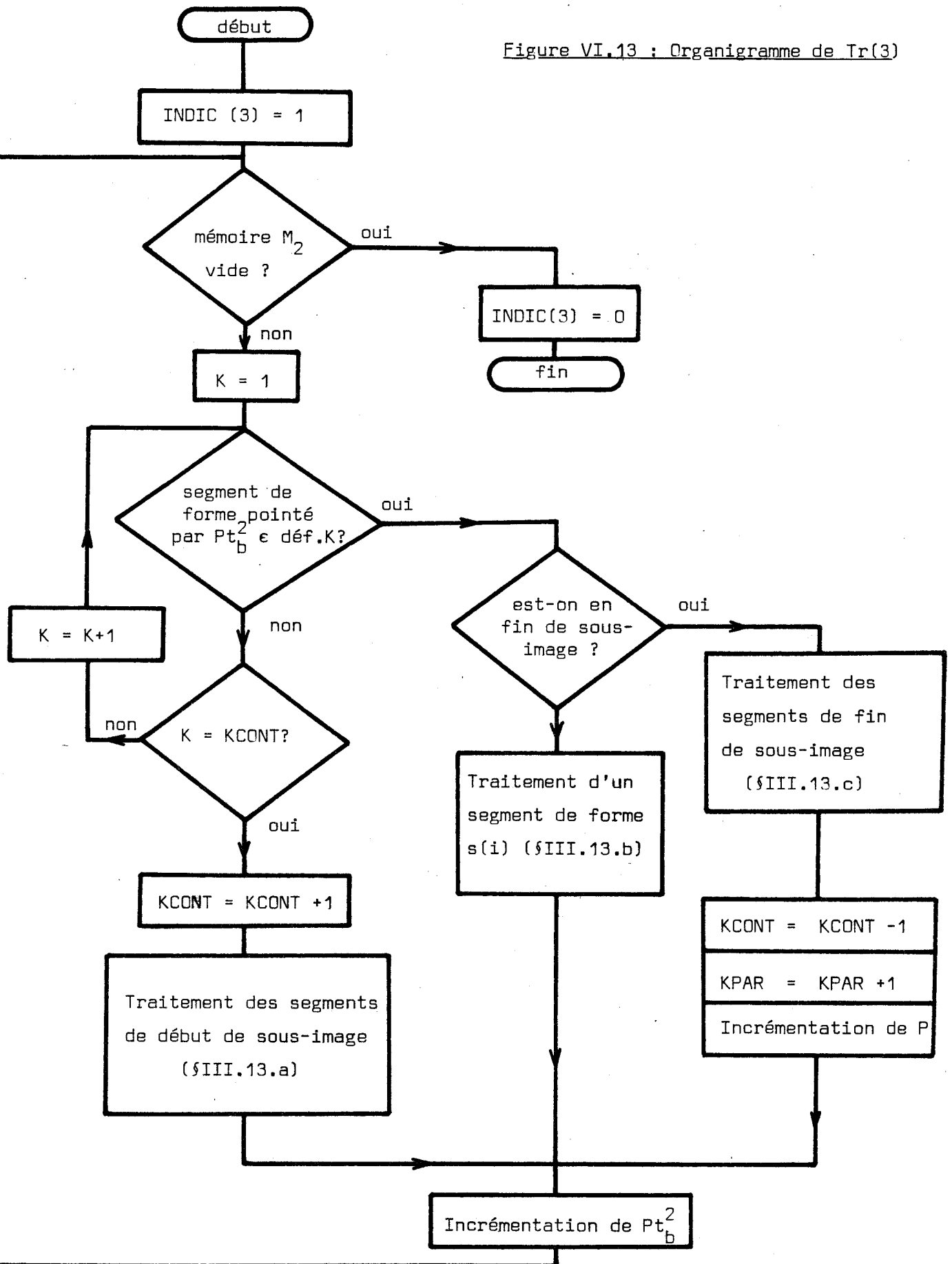
- * REF : référence de la forme.
- * j_d^i : position de la première détection de la forme.
- * C_g^k, C_d^k : associées aux positions C_d^{k+1} , ces valeurs permettent le calcul des éléments des chaînes de contour entre les segments de forme $s(k)$ et $s(k+1)$.
- * S_g^k, S_d^k permettent le lissage des seuils S_g^{k+1} et S_d^{k+1} (voir Chapitre III)
- * Pt_g et Pt_d sont les pointeurs indiquant les adresses de la mémoire M_3 où sont rangés les éléments de contour gauches et droits calculés pour le segment de forme $s(k+1)$.

VI.4.3. Organigramme général de la transformation Tr(3)

L'organigramme de la figure VI.13 regroupe les principales phases de la recherche de contour :

- si un segment de forme n'appartient pas à l'une des kCONT formes en cours de traitement, la procédure spécifique aux segments de forme du début des sous-images est utilisée, puis kCONT est incrémenté.
- si un segment $s(i)$ appartient à l'une des formes en cours de détection de contour, soit le traitement d'un segment de forme $s(i)$, soit le traitement de fin de sous-image est employé. Dans ce dernier cas, le nombre de formes en cours de traitement Tr(3) est décrémenté ($kCONT = kCONT - 1$) et le nombre de formes à paramétrer est incrémenté ($kPAR = kPAR + 1$).

Figure VI.13 : Organigramme de Tr(3)



VI.4.4. Transformation Tr(4)

La transposition logicielle de la paramétrisation et de la reconnaissance des formes ne présente aucune particularité susceptible d'être incluse dans cette partie. En effet, la transformation Tr(4) est une suite d'opérations indépendantes les unes des autres.

Citons néanmoins que le calcul des paramètres géométriques des formes ne s'effectue pas directement sur les informations stockées dans la mémoire M_3 . Il faut réorganiser les deux chaînes de contour droit et gauche pour ne former qu'une seule chaîne. A cette fin, une table est créée dont la taille est suffisante pour contenir tous les éléments de contour du plus grand défaut possible.

La figure VI.14 montre que la chaîne globale s'obtient en inversant l'ordre des éléments de la chaîne gauche.

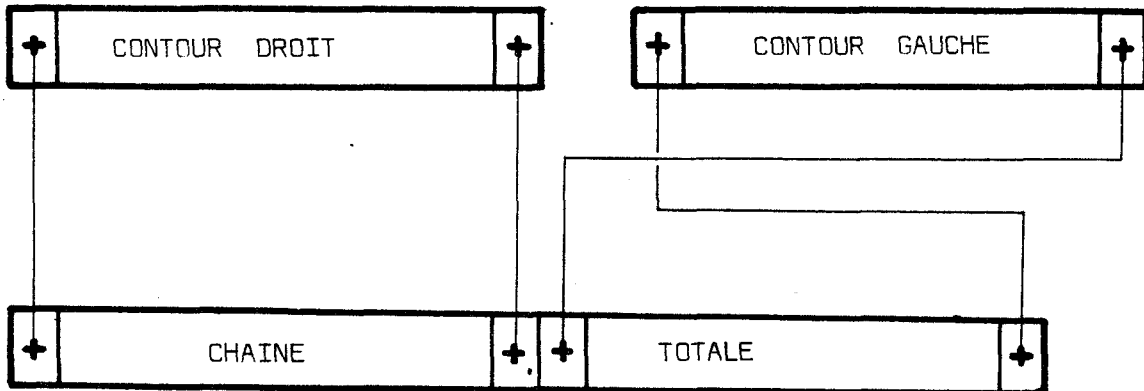


Figure VI.14 : Constitution de la chaîne totale de contour

VI.5. Conclusions

La décomposition de la procédure globale d'acquisition et de traitement d'images en tâches de degré de priorité différent a permis d'une part une implantation en temps réel et d'autre part la minimisation de la taille mémoire du calculateur . Une partie du chapitre suivant met en évidence que l'accroissement des performances du prototype en vitesse d'analyse est en partie lié à l'accroissement de la taille mémoire du système.

C H A P I T R E VII

- PERFORMANCES DU SYSTEME ET PERSPECTIVES D'AMELIORATION

Le système d'acquisition, de traitement et de reconnaissance d'images, élaboré dans une perspective industrielle autour d'un microprocesseur et avec des moyens informatiques volontairement limités, présente, en temps réel, des performances relativement intéressantes mais qui sont susceptibles d'amélioration. Le but de ce chapitre est d'une part d'apprécier les performances du système actuel et d'autre part de proposer certaines améliorations possibles.

VII.1. Performances générales du prototype

VII.1.1. Performances de rapidité

VII.1.1.1. Influence de l'hétérogénéité du contexte

Les performances de vitesse sont pratiquement indépendantes de la densité d'impuretés du peigné; en effet, la fréquence d'apparition des défauts est très faible (un défaut toutes les 6 secondes en moyenne) et par ailleurs la procédure de traitement complet, c'est-à-dire de l'acquisition à la reconnaissance, est très rapide. Ainsi, pour une paille de 4 mm, la reconnaissance est obtenue 0,3 seconde après le début de la détection.

En réalité, la vitesse d'analyse est essentiellement limitée par le traitement du nombre important de fausses détections. Suivant la qualité de présentation du voile de laine, le nombre moyen de détections par ligne varie de 1 à 1,5.

La validation d'une détection nécessite 4ms, l'analyse d'une ligne de l'image s'effectue en 10 ms : en conséquence, 40 à 60% du temps est inoccupé. A priori, il est donc possible d'augmenter la vitesse du peigné de façon à réduire ce temps inoccupé.

En fait, ce gain de vitesse se fait au détriment de la place mémoire. En effet, l'image fugitive composée de 8 lignes est un tampon permettant de faire face aux fortes densités de détections; pour accroître la vitesse d'analyse, il faudrait donc augmenter considérablement la taille de cette mémoire.

VII.1.1.2. Capacité de traitement du prototype

Pour éviter tout risque de saturation lors d'une forte densité de défauts, la mémoire contenant les données utiles au traitement est organisée en trois files d'attente (M_1 , M_2 , M_3) et une mémoire tampon (image fugitive).

Sur les 12K octets constituant le volume de la mémoire, la mémoire image occupe les deux tiers, la file d'attente M_2 un quart et le reste (1K octet) correspondant à M_1 et M_3 .

Il est intéressant de noter que la mémoire image et la file d'attente M_2 (mémoire des sous-images) ont un rôle similaire : la première est un tampon lors des fortes densités de détections et la seconde est également un tampon, lors des fortes densités d'impuretés. A titre d'exemple, M_2 mémorise jusqu'à 4 défauts de 6 mm, ceci dans le cas le plus défavorable, c'est-à-dire quand les impuretés apparaissent simultanément sur les mêmes lignes de l'image. Dans le cas où ces défauts se présentent séquentiellement (aucune ligne commune), la capacité de M_2 est accrue d'un facteur 6. Ce gain est la conséquence de la gestion de l'acquisition des sous-images et de la détection de contour, par interruptions logicielles. En effet, lorsqu'une sous-image est en cours d'acquisition, la recherche de contour de celle-ci peut être entamée, ce qui réduit la partie de sous image mémorisée dans M_2 (figure VII.1).

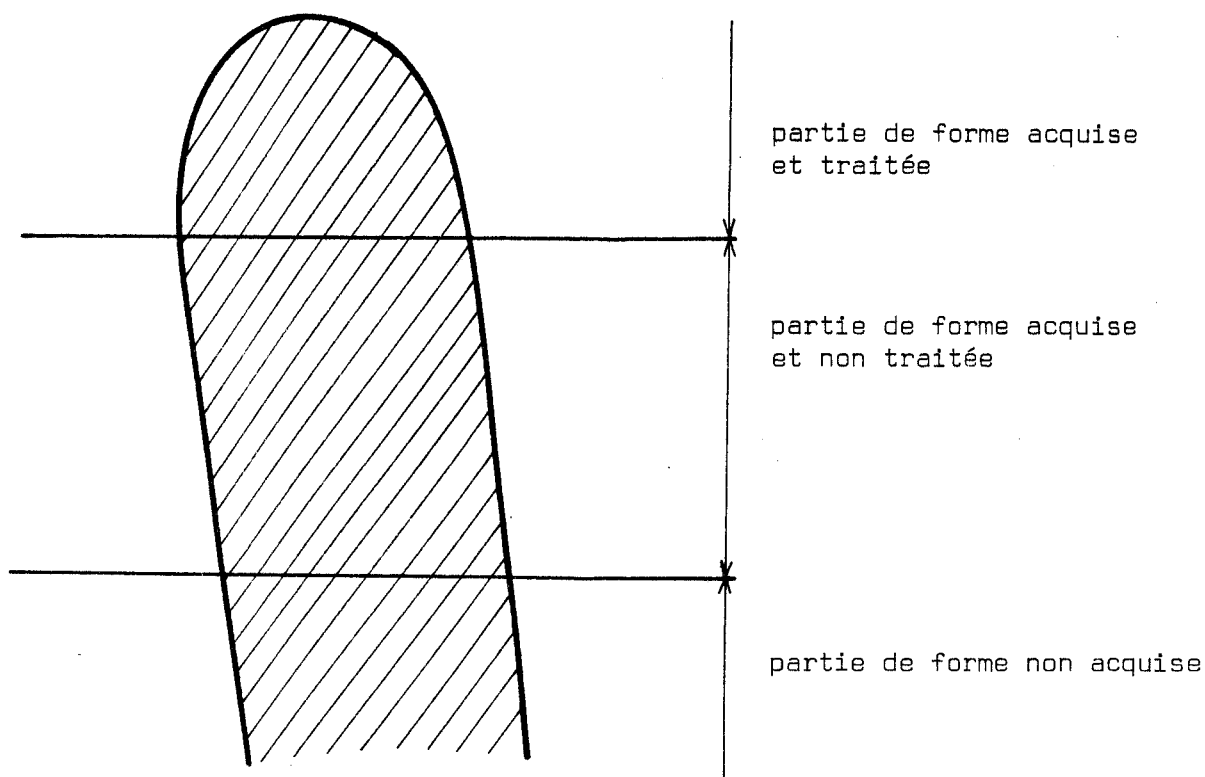


Figure VII.1 : Acquisition et traitement d'une sous-image quasi-simultanés

VII.1.2. Limites de sensibilité

La détection des formes de très faible largeur ou faiblement contrastées est liée à différents paramètres dont nous allons examiner l'influence :

- choix du seuil de détection par voie matérielle : T_s

Nous avons vu au chapitre II (§II.3.2.) que la détection, par voie matérielle, des zones de l'image à fort contraste s'obtient par comparaison de l'image $T(i,j)$ à une image filtrée $T^*(i,j)$ correspondant au contexte. Dans ce but, le signal représentant $T^*(i,j) - T(i,j)$ est comparé à un seuil fixe T_s . Or, la diminution de ce seuil permet d'accroître la sensibilité et donc de détecter des contrastes faibles. T_s est donc choisi pour que la forme la plus faiblement contrastée soit détectée.

- choix des seuils P_s et C_s

Après calcul des paramètres P_1 , P_2 , C_1 , C_2 (§II.3.3.), une détection est validée si les pentes et contrastes sont supérieurs respectivement à P_s et C_s .

Ces seuils sont choisis de façon à ce que les détections de formes faiblement contrastées soient validées.

Au cours des essais visant à fixer p_s et c_s ; nous avons pu faire quelques constatations :

- la validation des détections relatives à des boutons est liée principalement au choix de P_s . En effet, les boutons se caractérisent par des contrastes C_1 , C_2 relativement élevés, les pentes P_1 , P_2 étant faibles.

- Au contraire, les matières végétales se caractérisent généralement par des pentes élevées et des contrastes parfois faibles.

La figure VII.2. montre les résultats obtenus pour une paille de très faible largeur (0,17 mm), dimension inférieure au pas d'échantillonnage de l'image (0,2 mm). Cette figure fait apparaître que la largeur de la paille est nulle pour, pratiquement, toutes les lignes de la sous-image. Ce résultat est la conséquence de l'échantillonnage de l'image : un point de l'image (i,j) représente un élément de surface de 0,2 mm x 0,2 mm centré sur ce point. Par conséquent, lorsque les frontières C_d^i et C_d^i d'un segment de forme sont confondues, la largeur de la forme est en fait, égale au pas d'échantillonnage.

```

70 68 71 69 69 69 72 72 72 71 70 72
70 68 68 67 65 64 63 71 72 72 70 72
70 69 67 66 64 53 62 63 72 72 68 69
71 70 68 67 64 53 61 63 73 73 69 70
70 71 70 68 64 53 62 63 73 74 70 71
71 72 70 68 64 52 62 63 73 73 70 70
71 70 68 68 64 64 63 69 72 71 69 70
72 68 68 68 64 55 63 68 71 70 71 70
72 68 69 69 64 55 63 68 70 71 73 71
71 70 69 69 64 56 63 67 71 73 72 70
72 70 69 69 64 57 63 67 72 73 70 70
73 72 70 68 67 64 56 63 63 72 74 70
73 73 70 67 64 45 56 63 63 72 73 69
73 74 74 69 66 64 46 59 63 70 73 70
73 75 74 69 65 64 47 62 63 71 73 69
75 75 74 70 67 64 51 63 63 72 72 69
76 73 71 67 64 51 62 63 72 72 70 72
77 74 71 69 64 50 60 63 71 72 72 74
76 72 69 64 48 58 63 69 72 72 74 74
75 71 70 64 64 55 63 68 72 71 74 75
75 72 69 68 64 64 57 63 70 72 72 74
72 71 70 68 64 59 63 70 72 72 72 73
75 74 73 71 66 67 63 69 72 71 72 72
73 74 72 73 71 73 71 69 72 71 71 72
    
```

S _g	S _d	P ₁	P ₂
58	62	88	50
58	62	88	53
58	62	88	53
58	62	96	56
58	62	96	56
58	62	96	52
60	61	56	52
60	61	56	52
60	61	56	52
60	61	56	52
60	61	56	52
57	59	52	54
56	58	44	64
55	58	36	64
56	59	104	56
56	60	104	56
57	60	76	56
57	59	84	56
57	59	84	52
57	59	84	52
57	59	84	52

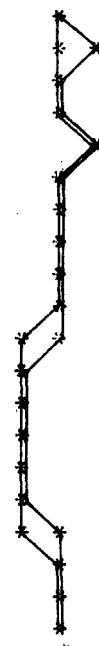
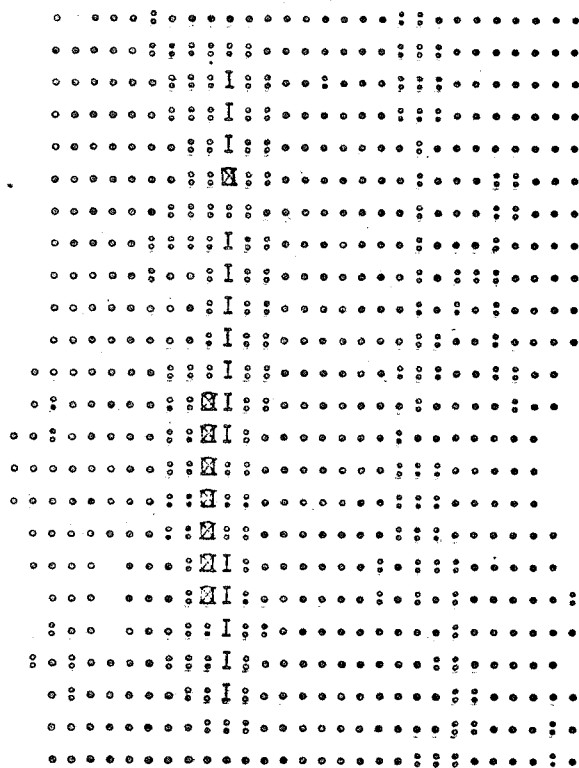


Figure VII.2 : Sous-image d'une paille très fine

VII.1.3. Reproductibilité des résultats

VII.1.3.1. Etude pour une paille et un bouton

L'étude de reproductibilité des résultats consiste à analyser une même forme dans différentes situations (position, orientation). Le changement de position des formes dans le peigné permet d'analyser l'influence du changement de l'intensité lumineuse du contexte.

Les résultats des essais réalisés pour une paille et un bouton sont portés sur les figures VII.3 et VII.4. A chaque contour de forme, nous avons associé la longueur L , la surface S , le paramètre représentant la compacité P^2/s .

Chaque forme a pu être reconnue sans erreur dans tous les cas considérés, car pour la paille $P^2/s > 32$ et pour le bouton $P^2/s < 15$.

VII.1.3.2. Etude comparative sur boutons.

Si la première série d'essais permettait de tester la qualité de l'acquisition vis à vis des problèmes de rotation des formes, cette seconde série vise un double objectif :

* comparaison entre l'analyse visuelle et les résultats du système d'acquisition.

* sensibilité au bruit des procédures d'acquisition et de traitement.

Pour répondre à ce double but, les essais ont été menés en trois étapes :

1. Acquisition et traitement des images d'un bouton dans son contexte (peigné).
2. Même opération hors contexte, le bouton étant placé sur un support plastique transparent (contexte d'intensité constante).
3. Observation au microscope du même bouton par plusieurs opérateurs.

Les mesures ont porté sur 9 boutons dont les diamètres sont voisins de la limite de prise en compte (1 mm). Pour effectuer une étude statistique des résultats, chaque bouton a subi 15 tests, ce qui représente un total de 270 mesures. Les impuretés ont été présentées dans le peigné dans toutes les configurations possibles : présentation sur fond de transparence variable, sur les bords du peigné, et ce pour diverses orientations des boutons.

Les résultats des essais sont présentés dans les deux tableaux de la figure VII.5 . Pour chaque bouton, les moyennes des longueurs et surfaces sont portées dans ces tableaux. Pour permettre une étude comparative des mesures avec et sans peigné, les écarts-type (E.C) et coefficient de variation (C.V) sont calculés .

L'étude de ces résultats nous a permis de dégager certaines constatations :

- Quelles que soient les conditions de mesure, c'est-à-dire avec et sans peigné, le coefficient de variation de la surface (S) est beaucoup plus élevé que celui du diamètre maximal (L). Il semble donc que la mesure du diamètre maximal soit plus représentative que la mesure de la surface.

- Les résultats sont évidemment plus dispersés lorsque les boutons sont placés sur le peigné (contexte bruité). En effet, le coefficient de variation moyen de S varie de 8,95% à 17,5 % alors que le coefficient de variation moyen de L varie certes dans le même sens, mais dans des proportions moindres : 6,33% à 8,85%.

La dispersion des résultats peut paraître élevée, mais il faut cependant tenir compte du fait que les dimensions des impuretés ne sont que 5 à 6 fois supérieures au pas d'échantillonnage. Les conditions de mesure sont en partie la cause de cette dispersion : il est difficile de déplacer les boutons entre chaque mesure, sans modifier leurs caractéristiques dimensionnelles.

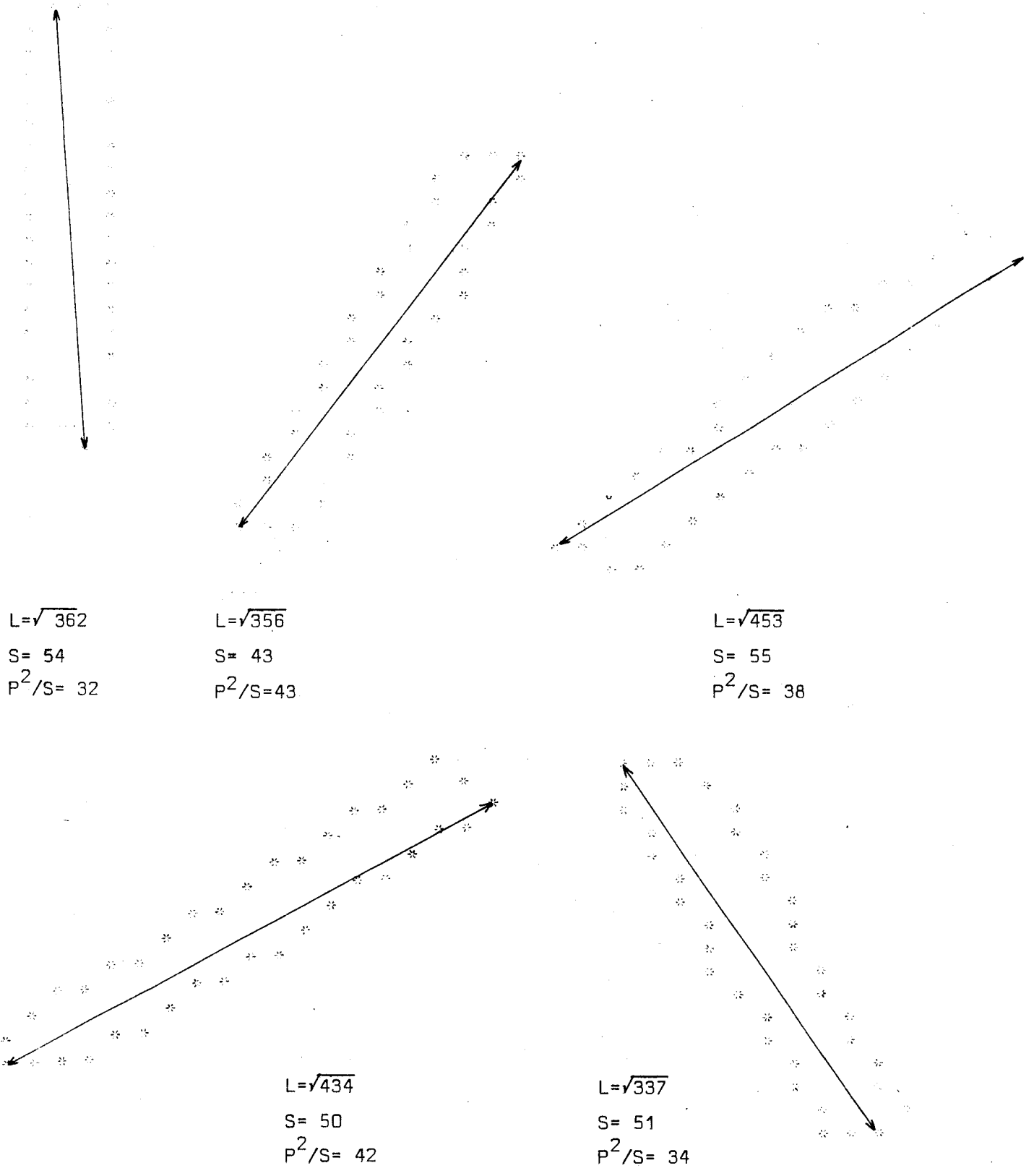
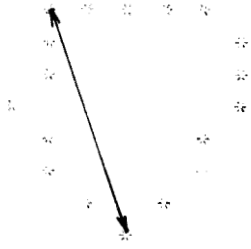
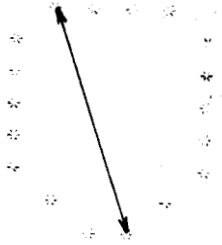


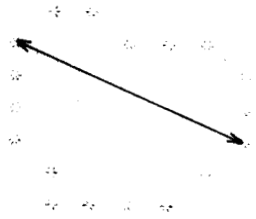
Figure VII,3 : Résultats pour une paille



$L = \sqrt{53}$
 $S = 28$
 $P^2/S = 15$



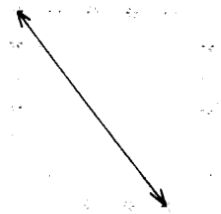
$L = \sqrt{53}$
 $S = 30$
 $P^2/S = 13$



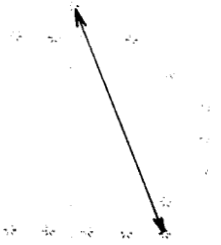
$L = \sqrt{45}$
 $S = 28$
 $P^2/S = 14$



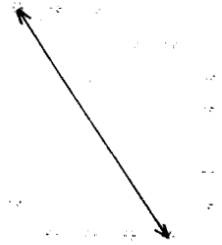
$L = \sqrt{53}$
 $S = 30$
 $P^2/S = 13$



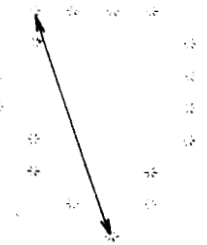
$L = \sqrt{45}$
 $S = 27$
 $P^2/S = 14$



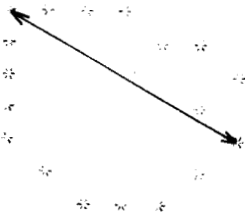
$L = \sqrt{53}$
 $S = 34$
 $P^2/S = 13$



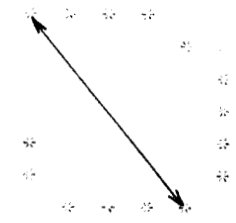
$L = \sqrt{52}$
 $S = 27$
 $P^2/S = 13$



$L = \sqrt{53}$
 $S = 24$
 $P^2/S = 15$



$L = \sqrt{53}$
 $S = 28$
 $P^2/S = 15$



$L = \sqrt{52}$
 $S = 30$
 $P^2/S = 13$

Figure VII.4 :
Résultats pour un bouton

Ref	surface			longueur			$\emptyset=L$ mm	$\emptyset=f(s)$ mm	\emptyset mesuré mm
	S _{mm}	E.C. _{mm}	C.V%	L _{mm}	E.C. _{mm}	C.V%			
B ₁	1,132	0,153	13,6	1,41	0,89	6,38	1,41	1,20	1,37
B ₂	1,008	0,161	16,04	1,32	0,95	7,2	1,32	1,13	1,17
B ₃	0,5118	0,113	22,2	1,04	0,121	11,7	1,04	0,80	0,86
B ₄	1,141	0,202	17,7	1,37	0,121	8,9	1,37	1,20	1,31
B ₅	1,063	0,206	19,4	1,39	0,134	9,7	1,39	1,16	1,21
B ₆	1,201	0,181	15,1	1,45	0,134	9,3	1,45	1,23	1,31
B ₇	1,126	0,203	18,1	1,42	0,150	10,6	1,42	1,19	1,17
B ₈	1,201	0,198	16,5	1,72	0,106	6,2	1,72	1,23	1,18
B ₉	1,663	0,307	18,5	1,73	0,167	9,7	1,73	1,45	1,36

Essais avec peigné

Ref	surface			longueur			$\emptyset=L$ mm	$\emptyset=f(s)$ mm	\emptyset mes. mm
	S _{mm²}	E.C. _{mm²}	C.V%	L _{mm}	E.C. _{mm}	C.V%			
B ₁	1,100	0,0688	6,26	1,39	0,058	4,2	1,39	1,18	1,37
B ₂	0,816	0,0799	9,8	1,18	0,061	5,2	1,18	1,01	1,18
B ₃	0,453	0,0507	11,2	0,85	0,085	10,0	0,85	0,76	0,86
B ₄	0,998	0,0798	8,0	1,30	0,070	5,45	1,30	1,12	1,130
B ₅	0,811	0,0827	10,2	1,18	0,079	6,7	1,18	1,01	1,18
B ₆	1,087	0,0793	7,3	1,41	0,115	8,2	1,41	1,16	1,31
B ₇	0,743	0,0676	9,1	1,20	0,046	3,9	1,20	0,97	1,17
B ₈	0,843	0,0918	10,9	1,31	0,098	7,5	1,31	1,03	1,18
B ₉	1,262	0,0984	7,8	1,48	0,087	5,9	1,48	1,26	1,36

Essais sans peigné

$\emptyset=L$: longueur maximale (mm)
 $\emptyset=f(s) = \sqrt{\frac{4S}{\pi}}$ (diamètre) (mm)
 \emptyset mesuré = mesure au microscope (mm)

Figure VII.5 : Tableaux de résultats pour des boutons.

VII.2. Perspectives d'améliorations et de généralisation

VII.2.1. Acquisition de formes quelconques

VI.2.2.1. Décomposition d'une forme en parties élémentaires

Dans la procédure d'acquisition présentée au chapitre II, une forme est supposée définie par un seul segment de forme $s(i)$ par ligne. La majorité des impuretés rencontrées dans notre application, vérifie cette hypothèse. Cependant certaines pailles sont en partie cassées dans le sens de leur longueur; cette division du corps en deux parties provoque plusieurs détections par ligne. L'acquisition des sous-images des branches d'une même forme est alors nécessaire pour connaître avec exactitude la longueur maximale (L) de la forme complète .

Tous les cas de séparation de forme en plusieurs branches sont résolus, si l'on est capable de traiter les deux cas suivants (figure VII.6).

- La forme se présente initialement en une seule partie, puis se sépare en deux parties (cas 1)
- Initialement, deux formes distinctes sont détectées, les deux branches se rejoignant pour finalement former un seul corps (cas 2).

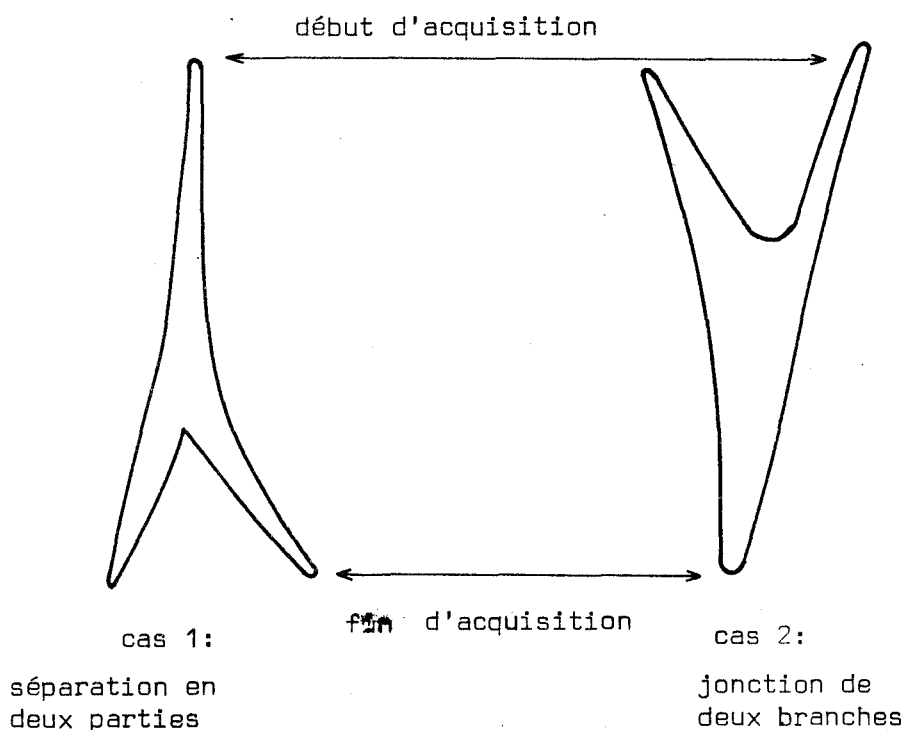
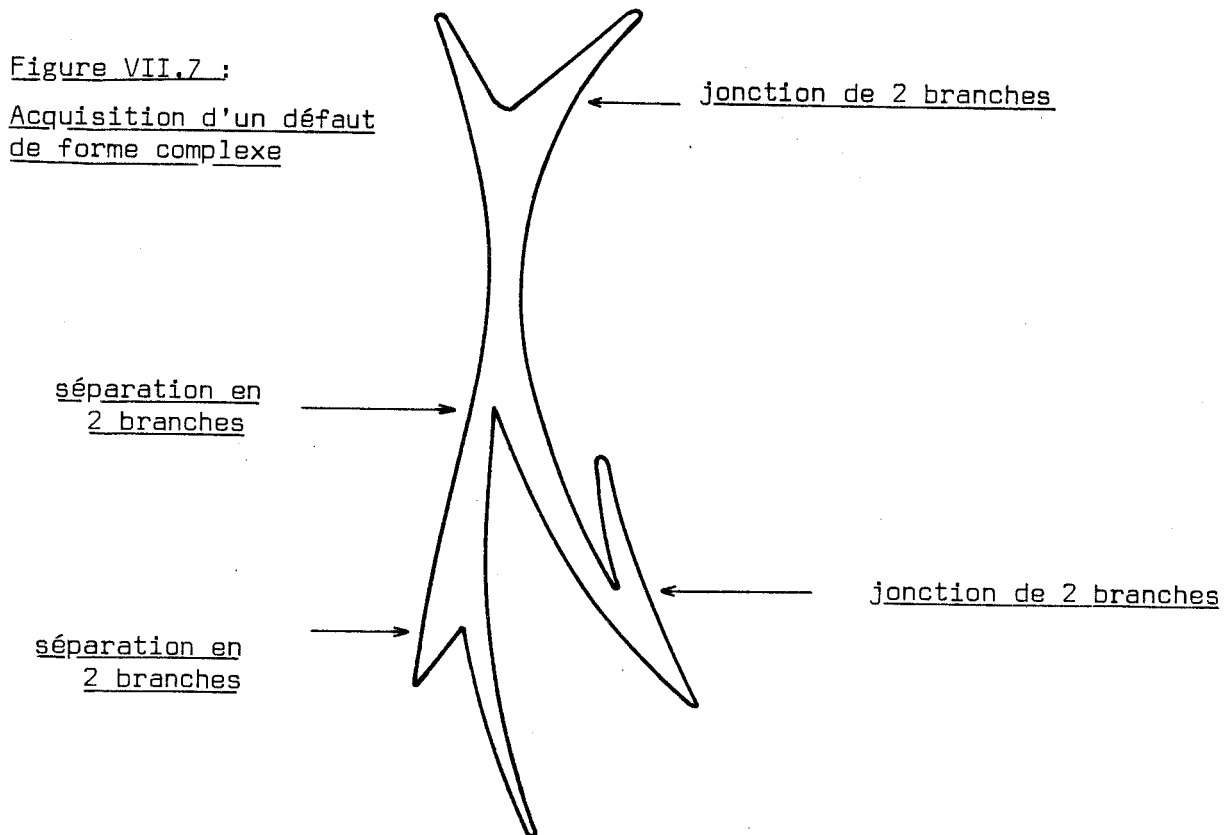


Figure VII.6 : Décomposition d'une forme en plusieurs branches

La figure VII.7 montre comment une forme quelconque peut se décomposer en parties élémentaires : jonction de deux branches et séparation du corps en deux branches .



VII.2.1.2. Modification de la procédure d'acquisition

VII.2.1.2.1. Cas de la séparation de la forme en deux branches

Pour expliciter la procédure employée, développons les différentes phases d'acquisition à partir d'un exemple (figure VII.8) :

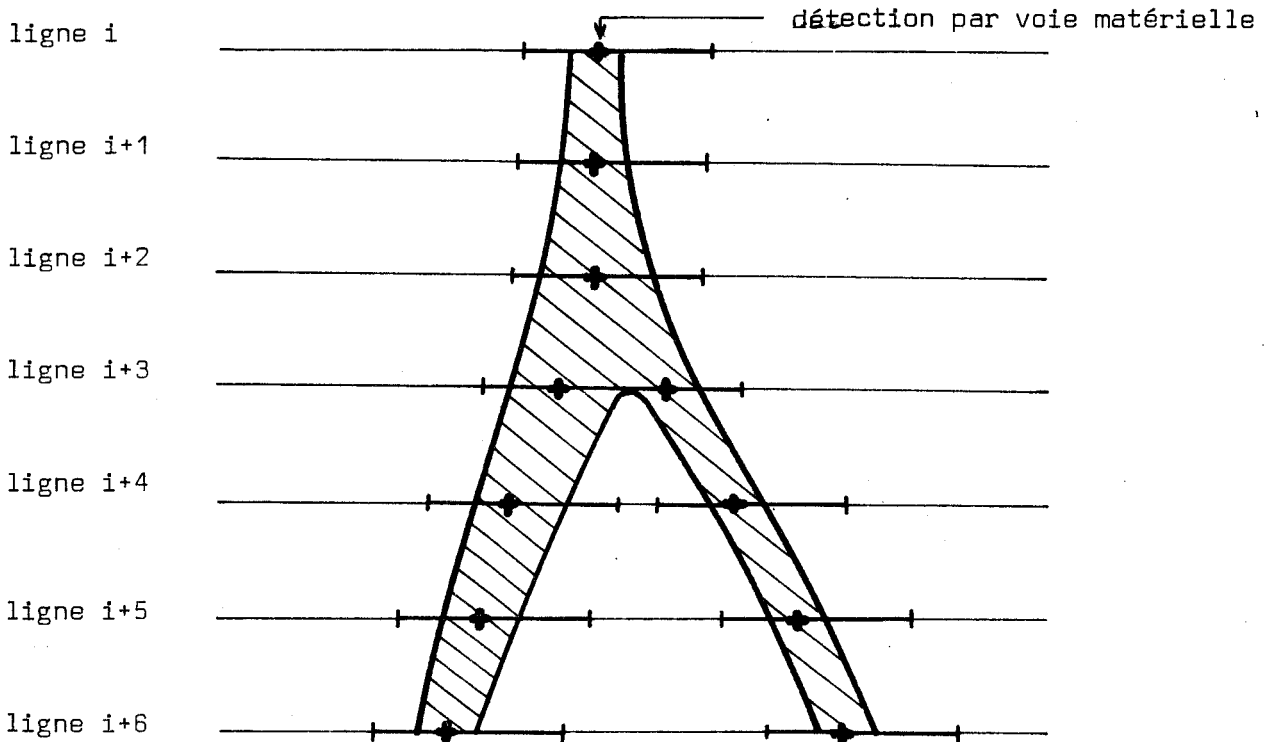


Figure VII.8 : Acquisition d'une sous-image dans le cas de séparation en deux branches

- lignes i à i+2.

La forme étant définie par une seule détection par ligne, la procédure d'acquisition de la sous-image reste inchangée (chapitre II).

- ligne i+3.

Les deux détections $(j_d^{i+3})_1$ et $(j_d^{i+3})_2$ appartiennent à la même forme, car la relation de "connexité à priori" est vérifiée (§II.4.1) :

$$(j_d^{i+3})_1 \text{ et } (j_d^{i+3})_2 \in [j_d^{i+2} - \Delta a, j_d^{i+2} + \Delta a]$$

Dans ce cas, le segment de forme acquis est l'ensemble suivant :

$$\{T(i, j) \text{ , } j \in [(j_d^{i+3})_1 - \Delta 1, (j_d^{i+3})_2 + \Delta 2]\}$$

- ligne i+4 à i+6.

De façon à inclure les détections $(j_d^1)_1$ et $(j_d^1)_2$ à la même forme, la relation de "connexité a priori" doit être testée à partir des détections $(j_d^{1-1})_1$ et $(j_d^{1-2})_2$. Les segments de forme acquis sur une même ligne l sont dissociés si $(j_d^1)_1 + \Delta 2 < (j_d^1)_2 - \Delta 1$. Dans le cas contraire le segment acquis est :

$$\{T(i,j) , j \in [(j_d^1)_1 - \Delta 1 , (j_d^1)_2 + \Delta 2] \} .$$

VII.2.1.2.2. Cas de la jonction de deux branches

Comme pour le cas précédent , établissons la procédure à partir d'un exemple (figure VII.9).

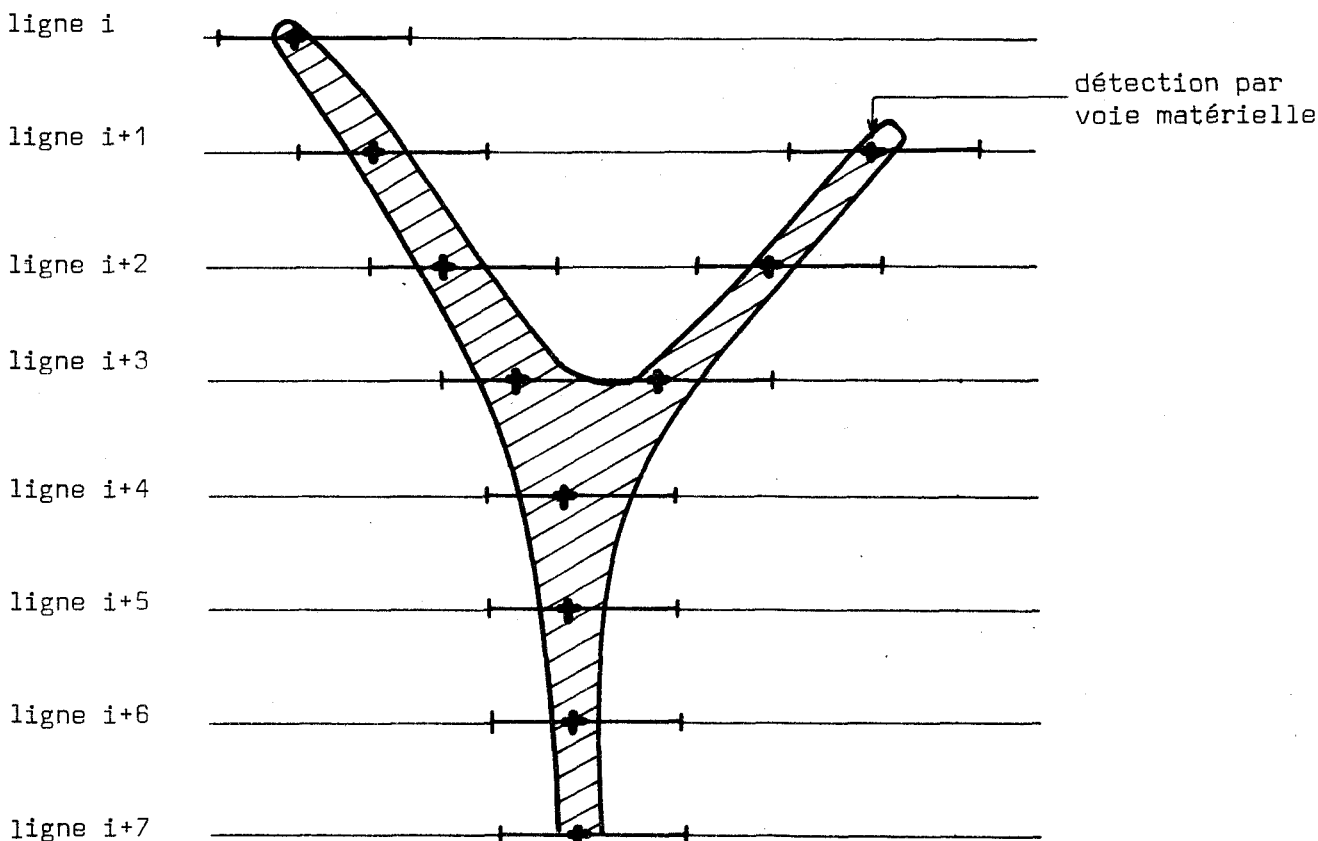


Figure VII.9: Acquisition d'une sous-image dans le cas de jonction de deux branches

- ligne i à i+3.

Sur ces lignes, il n'est pas encore possible de savoir si les détections appartiennent à une même forme. Donc, à priori, deux sous-images sont en cours d'acquisition.

- ligne i+4.

La détection (j_d^{i+4}) vérifie la relation de "connexité à priori", à la fois, pour les deux sous-images en cours d'acquisition. Les sous-images sont donc incluses à la même forme.

Cette réunion des deux branches suppose donc que la relation de "connexité à priori" soit testée pour chacune des formes en cours d'acquisition.

- ligne i+5,...

L'acquisition de la sous-image se déroule suivant la procédure établie au chapitre II.

VII.2.1.3. Modifications de la recherche de contour.

La détection de contour de formes possédant plusieurs branches nécessite quelques précautions . En effet, lors de l'acquisition il n'est pas possible de vérifier si deux branches sont réellement jointives ou si deux formes sont très proches (figure VII.10).

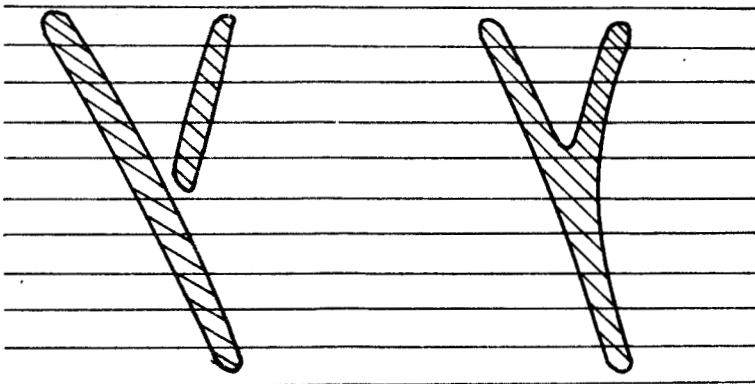


Figure VII.10: Cas identiques vis à vis de l'acquisition

Lorsque deux détections d'une même forme sur une ligne i sont proches, le nombre de frontières peut être égal à 2 ou 4 (figure VII.11).

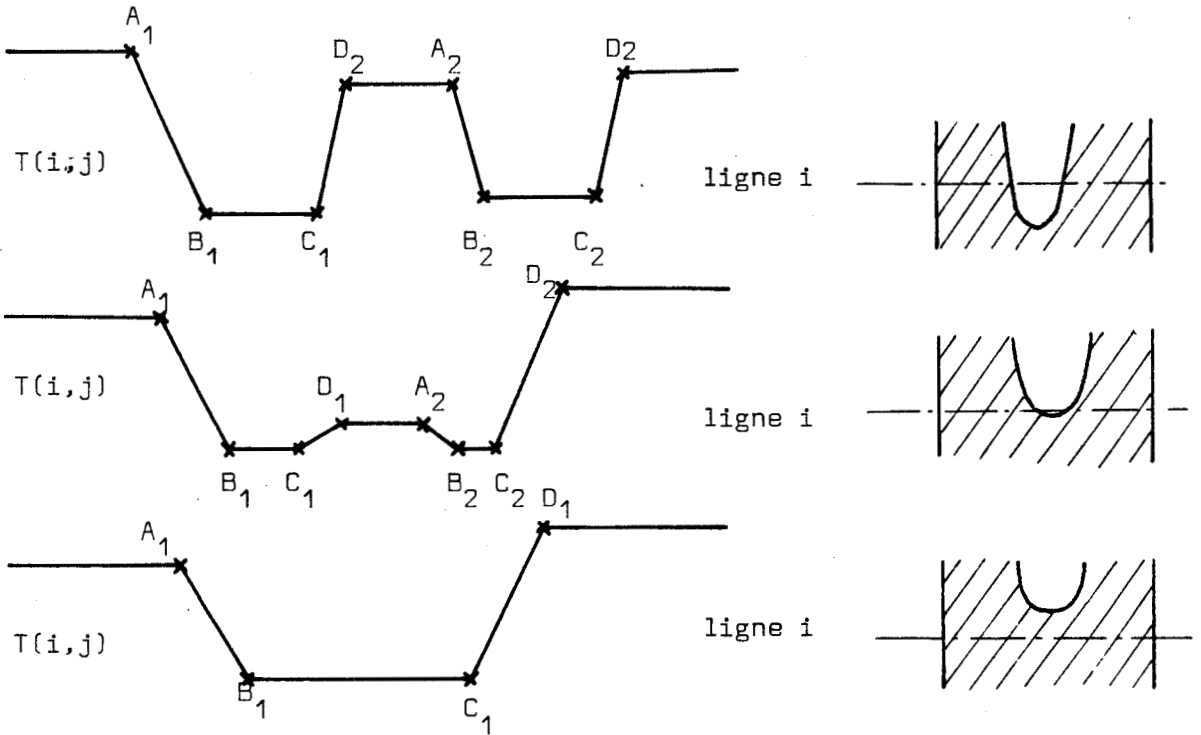


Figure VII.11: Recherche des frontières dans le cas de plusieurs branches.

La méthode de recherche de contour employée dans le cas de deux frontières (§ III) peut s'appliquer moyennant quelques opérations supplémentaires. Les différentes phases du traitement d'un segment de forme sont les suivantes :

- A partir de j_{\min} , recherche des points A_1, B_1, C_1, D_1 .
- Recherche d'un minimum local dans le segment au delà des points A et D.
- Recherche des points A_2, B_2, C_2, D_2 à partir de ce minimum local.

Si ces derniers points n'existent pas, nous retrouvons le cas C de la figure VII.11. Le cas b est obtenu lorsque les contrastes C_{g1} , C_{d2} de pentes P_{g2} , P_{d2} sont trop faibles.

Il est facile d'étendre cette méthode à la recherche de $2 \times n$ frontières dans le cas de présence de n branches très proches.

VII.2.2. Symétrisation de la recherche de contour

Les résultats des mesures sur boutons (§ VII.1.3.2.) ont mis en évidence l'influence du peigné, donc d'un contexte bruité, sur la dispersion des paramètres : surface et longueur . Cette dispersion peut être réduite par une amélioration de la recherche de contour.

Le tracé de l'évolution des seuils S_g et S_d le long d'une forme, c'est-à-dire en fonction de l'indice de ligne i , permet de remarquer un relèvement en début et fin de forme (premières et dernières lignes) ; la figure VII.12. montre quelques exemples d'évolution des seuils pour des boutons .

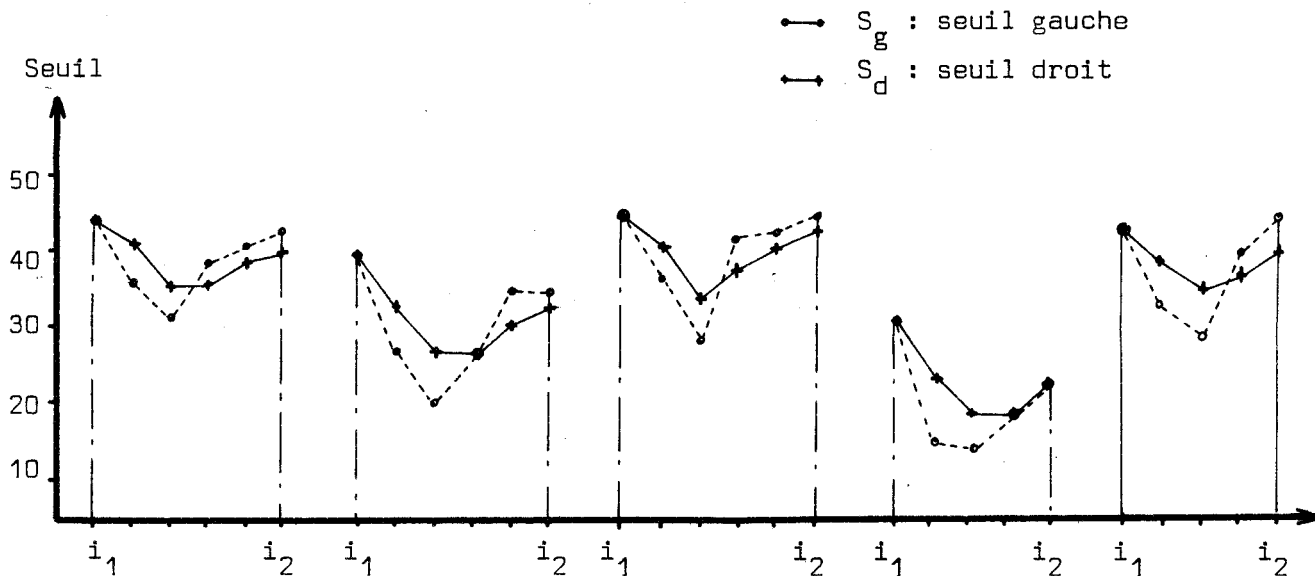


Figure VII.12 : Evolution de seuils pour quelques boutons

Le relèvement des seuils est inhérent à la méthode de recherche de contour : les premiers et derniers segments des sous-images présentent généralement des contrastes très faibles ; par conséquent les seuils sont élevés dans ces zones . La figure VII.12 fait apparaître des relèvements de seuils plus importants en début qu'en fin de sous-image. Cette particularité est due à la nature asymétrique de la détection de contour : l'application d'un filtre asymétrique $S_i^* = a S_i + b S_{i-1}^*$ a pour conséquence la conservation de seuils élevés en début de défaut, alors que sur les derniers segments de forme, l'augmentation des seuils est partiellement filtrée.

La modification proposée a pour but de rendre la procédure tout a fait symétrique. La figure VII.13 montre la décomposition de la procédure en trois étapes :

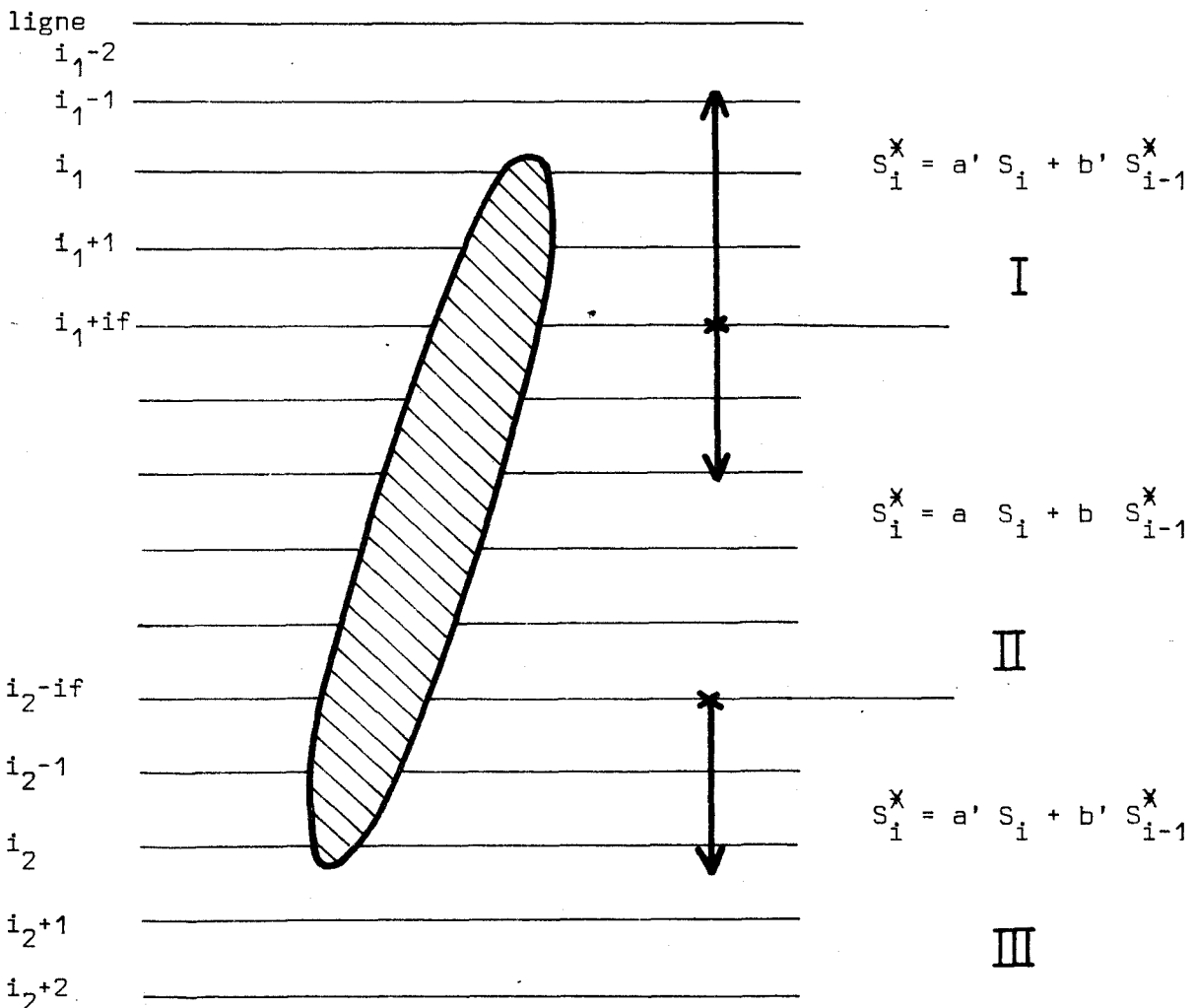


Figure VII.13 : Modification de la recherche de contour

- Région I : de la ligne i_1-2 à i_1+if

L'investigation de la sous-image débute à la ligne i_1+if et se termine à la ligne i_1-2 . Sur cette partie de la forme, le filtrage des seuils est important, ce qui diminue l'influence des premières lignes :
 $S_i^* = a' S_i + b' S_{i-1}^*$. Des résultats satisfaisants ont été obtenus pour $if = 2$; $a' = 0,2$; $b' = 0,8$.

- Région II : de la ligne i_1+if+1 à i_2-if-1

La méthode de détection de contour explicitée dans le chapitre III est appliquée à ces lignes.

- Région III : de la ligne i_2-if à i_2+2

Le traitement de cette région est identique au traitement appliqué à la région I.

CONCLUSION GENERALE

Avant d'envisager le développement industriel d'un appareil d'inspection automatique des peignés de laine , nous devons apporter, néanmoins, certaines améliorations :

- au niveau de la détection des impuretés :

L'extension de la procédure d'acquisition à des formes complexes permettrait d'éliminer les erreurs provoquées par la présence de pailles en partie cassées.

- au niveau de la reconnaissance :

L'identification des pailles et boutons est correctement résolue grâce à notre algorithme. Néanmoins, il faudrait poursuivre les essais pour évaluer correctement le taux de reconnaissance. De plus, nous n'avons pas envisagé, dans cet exposé, la reconnaissance des chardons dont la présence est, certes, très rare dans les peignés. Pour ce type de défaut, une analyse des contours de forme permettrait de détecter des aspérités, principales caractéristiques des chardons.

Dans ce mémoire, nous avons particulièrement traité le cas d'images bruitées, limitant par conséquent, les performances de rapidité du système. En effet, pour le problème d'inspection de la laine, la fréquence d'apparition des défauts est faible, alors que le nombre de fausses détections, éliminées par un algorithme, est très élevé. Dans le cas d'applications pour lesquelles le support des formes est plus homogène (tissu, par exemple), la vitesse d'analyse peut être considérablement augmentée.

Nous ne saurions conclure sans souligner que ce problème de détections de formes, qui ne présente aucune difficulté lors d'un examen visuel, se révèle très complexe lorsqu'on envisage une inspection automatique. Ainsi, une meilleure connaissance des phénomènes physiologiques du système oeil-cerveau permettrait une progression dans le domaine de la détection et reconnaissance de formes.

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

- |1| I. RIVAILLIER, B. COURTOIS :
Système d'identification rapide de caractères écrits sur microfilms.
Congrès de Rec. de formes AFCET - IRIA , Février 78, p. 612-622

- |2| E. HISDAL, N. CHRISTOPHERSEN, T. ERICSON, A LOKNETANGEN , S. VESTAL :
Structural recognition of hand - writing
3^{ie} Conf. Int. de Rec. de formes, nov. 76 , p. 140-143

- |3| A. CHAPELLE, G. LDRETTE, J.P. REQUIER, I. SERRA :
Reconnaissance automatique de caractères manuscrits bâton.
Congrès de Rec. de formes AFCET - IRIA, Fév. 78, p. 560-567

- |4| B. CARCENAC DE TORNE :
Identification des adresses postales
Congrès de Rec. de formes AFCET - IRIA , Fév. 78 p. 601-611.

- |5| C. BELLISSANT :
Un algorithme de segmentation de la parole.
Congrès de Rec. de formes AFCET - IRIA , Fév. 78 p. 475-488

- |6| T.D. GARVEY, J.M. TEMENBAUM :
Application of interactive scene analysis techniques to cartography
3^{ie} Conf. Int. de Rec. de formes, nov. 76, p. 213-217

- |7| G. NAGY :
Digital image-processing activities in remote sensing for earth
resources
Proc. IEEE, vol. 60 , Oct. 76 , p. 1177-1200

- |8| F. DURET :
Analyse d'images holographiques dentaires en vue de la commande des systèmes automatiques.
Sem. sur le traitement d'images bio-médicales de tous, mai 76
- |9| B. ZAVIDOVIQUE, G. STUMAN :
A biomedical image acquisition and processing system
3^{ie} Conf. Int. de rec. de formes, nov. 76, p. 360-366
- |10| A. DORINSON, G. GAILLAT :
Analyses de Scènes et élaboration de plans en robotique : une expérience sur mini-ordinateur.
Congrès de Rec. de formes AFCET - IRIA , Fév. 78 p. 127-135
- |11| L.S. COLES :
Nettoyage du sol par un robot
Congrès de Rec. de formes AFCET - IRIA , Fév. 78, p. 117-126
- |12| K. RAA, K. BLACK :
Type classification of finger prints : a syntactic approach
3^{ie} conf. Int. de Rec. de formes , nov. 76, p. 778-782
- |13| L.D. HARMAN, F.W. HUNT :
Automatic recognition of human face profiles.
3^{ie} Conf. Int. de Rec. de formes, nov. 76 p. 183-188
- |14| K.S. FU :
Syntactic methods in pattern recognition
Academic Press 1974
- |15| R.O. OUDA, P.E. HART :
Pattern Classification and Scene Analysis
Wiley- Interscience publication 1973

- |16| K.S. FU, A. ROSENFELD :
Pattern Recognition and Image Processing
IEEE Trans. on Computers, vol C.25, n°12, déc. 76, p. 1336-1346
- |17| R. JEANSOULIN :
Compression d'images numérisées utilisant la transformation de Walsh-Hadamart
Automatisme, mars Avril 76 , p. 98-101
- |18| G.E. LOWITZ :
Compression des données images par reconnaissance des formes et clustering
Congrès de Rec. de formes AFCET - IRIA , Fev. 78, p. 699-714
- |19| C. ROCHE , M. REBUFFET :
Méthodes de classification d'imagerie multispectrale.
Congrès de Rec. de formes AFCET - IRIA fév. 78, p. 715-729
- |20| G.E. LOWITZ :
Traitement orthogonal des images multispectrales
Automatisme, mars - avril 76 , p. 83-89
- |21| T. ASSEFI :
Modeling of the two - dimensional signals with application to image restoration
3^{ie} Conf. Int. de Rec. de formes, nov. 76, p. 696-700
- |22| P. MORTON, A. BRUEL :
Restauration d'images dégradées par des systèmes linéaires
Cong. de Rec. de formes AFCET - IRIA , fév. 78, p. 54-62
- |23| J. FLEURET :
Suppression de variations d'amplitude dans le spectre d'un signal : application à l'augmentation de contraste .
Cong. de Rec. de formes AFCET - IRIA , fev. 78, p. 534-541

- [24] L.G. ROBERTZ :
Machine perception of three dimensional solids.
Optical and Electr-optical information processing, 1965, p. 159-197
- [25] A. GUZMAN :
Decomposition of a visual scene into three dimensional bodies.
Proceedings of the AFIPS fall joint Conf., déc. 68, p. 291-304
- [26] G.L. CARAYANNOPOULOS, E.A. PATRICK :
An algorithm for segmentation of metaphase spreads
Journal of Pattern - Recognition, vol.8, 76, p. 151-161
- [27] S.L. HOROWITZ, T. PAVLIDIS :
Picture segmentation by a directed split and merge procedure.
2^{ie} Conf. Int. de Rec. des formes , Aout 74, p. 424-433
- [28] I.S.N. MURTHY, K.J. UDUPA :
A search algorithm for skelitonization of thick patterns
Computer graphics and image processing, 74, vol.3 p. 247-259
- [29] S. BEUCHER, F. MEYER :
Méthodes d'analyse de contrastes à l'analyseur des textures
Cong. de Rec. de formes AFCET - IRIA , fev. 78 p. 378-384
- [30] F.G. ARGILLAS, S. CASTAN :
Systèmes interactifs pour la recherche en traitement d'images et
leurs utilisations systématiques (SIRIUS)
Congrès de Rec. de formes AFCET - IRIA fev. 78, p. 872-830
- [31] J.P. CHIEZE, O. FAUGERAS :
Système interactif de traitement d'images
Congrès de Rec. de formes AFCET - IRIA , fev. 78, p. 836-840

- | 32 | F. GEORGES, P.L. WENDEL :
Système interactif d'analyse et de restitution d'images sur film
photographique
Congrès de Rec. de formes AFCET - IRIA , Fev. 78, p. 829-835
- | 33 | M.L. BAIRD :
An application of computer vision to automated I.C. chip manufacture
3^{ie} Conf. Int. de Rec. de formes, nov. 76 p. 3-5
- | 34 | P. SARAGA , D.R. SKOYLES :
An experimental visually controlled pick and place machine for
industry
3^{ie} Conf. Int. de Rec. de formes , nov. 76, p. 17-21.
- | 35 | W. FREI, C.C. CHEN :
Fast boundary detection : a generalisation and a new algorithm
IEEE Transactions on computers, Vol C26; n°10, oct.77,pp. 988-998.
- | 36 | M.H. HUECKEL :
An operator which locates edges in digitized pictures
J.Ass. Comput. Mack; vol 18, Janv 71, pp. 113-125
- | 37 | H. MAITRE, B. FAUST :
Détection de contours d'image dans un bus de reconnaissance des formes
Congrès AFCET-IRIA , Fév. 78, pp.456-465
- | 38 | NADLER :
L'application de l'opérateur de gradient vectoriel à la segmentation
d'images médicale
Séminaire sur le traitement d'images biomédicales, mai 76

- |39| O.H. BALLARD, J. SKLANSKY :
A ladder structured decision tree for recognizing tumors in chest radiographs.
IEEE Transactions on computers, vol 25, Mai 76, pp. 503-513
- |40| R.W. FHRICH, J. FOITH :
Representation for random waveforms by relational trees
Transactions on computers , vol C25, July 76, pp. 725-736
- |41| A. ROSENFELD :
Iterative methods in image analysis
Pattern recognition , vol 10, 78, pp. 181-187
- |42| H. FREEMAN :
On the encoding of arbitrary geometric configurations
IRE . Trans. Electron. Computers, june 61, pp. 260-268
- |43| H. FREEMAN :
Shape description via the use of critical points
Pattern recognition and image processing , vol 10, 78, pp. 159-166

TABLE DES MATIERES

Introduction générale	1
CHAPITRE I.	
- <i>SYSTEME DE RECONNAISSANCE DES FORMES</i>	
I.1. Domaines d'application de la reconnaissance des formes	3
I.2. Définition d'un système de reconnaissance	5
I.3. Traitement d'images	6
I.3.1. Définition d'une image digitale	6
I.3.2. Technique de traitement d'images	6
I.4. Matériel utilisé pour le traitement d'images	8
I.4.1. Utilisation des méthodes de traitement d'images en temps différé	8
I.4.2. Systèmes inter-actifs de traitement d'images	9
I.4.3. Notion de temps réel en traitement d'images	10
I.5. Conclusions	10
CHAPITRE II.	
- <i>ACQUISITION DES SOUS IMAGES EN TEMPS REEL</i>	
II.1. Présentation des images	
II.1.1. Traitement d'images en temps réel	12
II.1.2. Conditions d'analyse	13
II.2. Définition des segments de forme	15
II.2.1. Définition des sous-images dans le cas idéal	16
II.2.2. Définition des segments de forme dans un cas général	19
II.3. Acquisition des sous-images en temps réel	22
II.3.1. Principe de détection des formes	22
II.3.2. Prélocalisation par "voie matérielle" sous images	23
II.3.3. Détection des formes par voie logicielle	24
II.3.4. Exemples de recherche de segments de forme	27
II.4. Acquisition des sous-images par cadrage des formes	30
II.4.1. Relation de "connexité à priori"	30
II.4.2. Cadrage minimal des formes	30
II.4.3. Exemples d'acquisition de sous-images	32
II.5. Conclusion	35

CHAPITRE III.

- RECHERCHE DE CONTOUR DES FORMES

III.1. Généralités sur la recherche de contour	36
III.1.1. Techniques de recherche de contour	36
III.1.2. Utilisation de ces méthodes	39
III.2. Recherche de contour par seuils adaptatifs	39
III.2.1. Principe de la detection de contour	39
III.2.2. Recherche des frontières par seuillage adaptatif	41
III.2.3. Lissage des seuils adaptatifs	43
III.3. Adaptation aux sous-images acquises	43
III.4. Connexité des segments	47
III.4.1. Insuffisance de la "connexité à priori"	47
III.4.2. Amélioration de la définition de connexité des segments	48
III.5. Résultats de la recherche de contour	49
III.5.1. Détection de contour d'une forme compacte et d'une forme allongée	49
III.5.2. Histogrammes	50
III.5.3. Seuils adaptatifs	50
III.6. Conclusion	58

CHAPITRE IV.

- PARAMETRISATION DES FORMES

IV.1. Représentation des contours de forme par le codage de Freeman	59
IV.1.1. Connexité par "huit - voisinage"	60
IV.1.2. Représentation des contours grâce à des éléments directionnels	60
IV.1.3. Adaptation au mode séquentiel de recherche de contour	61
IV.2. Mesure de compacité	63
IV.2.1. Formulation de la compacité	63
IV.2.2. Mesure de périmètre	64
IV.2.3. Mesure de surface	64

IV.3. Dimensionnement d'une forme	67
IV.3.1. Définition de la longueur d'une forme	67
IV.3.2. Optimalité du résultat	70
IV.4. Conclusions	71

CHAPITRE V.

- REALISATION DU PROTOTYPE

V.1. Présentation du problème textile	72
V.2. Présentation du capteur optique	74
V.2.1. Dispositif de présentation de la laine	74
V.2.2. Choix et description du capteur	75
V.2.3. Optique et éclairage	77
V.3. L'unité de traitement informatique	77
V.3.1. Choix d'un calculateur	77
V.3.2. Principales caractéristiques du micro-calculateur	78
V.4. Interface caméra-calculateur	80
V.4.1. Synchronisation de la caméra	82
V.4.2. Interruptions	83
V.5. Acquisition d'image fugitive par accès direct mémoire	85
V.5.1. Organisation de la mémoire destinée aux images	85
V.5.2. Système à accès direct mémoire	86
V.5.3. Connexion de l'ADM par multiprocessing	88
V.6. Conclusion	92

CHAPITRE VI.

- IMPLICATIONS LOGICIELLES

VI.1. Décomposition de la procédure totale en tâches de niveau de priorité différent	93
VI.1.1. Implications du traitement en temps réel	93
VI.1.2. Description des tâches	94
VI.1.3. Imbrications des tâches	96
VI.1.4. Mémorisation des données intermédiaires	98
VI.1.5. Interruptions logicielles	101

VI.2. Etude de la transformation Tr(1)	104
VI.2.1. Description de la file d'attente M_1	104
VI.2.2. Gestion de la file d'attente M_1	106
VI.3. Etude de la transformation Tr(2)	107
VI.3.1. Organisation de la file d'attente M_2	107
VI.3.2. Organisation des secteurs de la file d'attente M_2	109
VI.3.3. Test d'appartenance d'un segment de forme à une sous-image en cours d'acquisition	109
VI.3.4. Organisation général de la transformation Tr(2)	111
VI.4. Etude de la transformation Tr(3),Tr(4)	114
VI.4.1. Organisation de la file d'attente M_3	114
VI.4.2. Fichier intermédiaire pour la recherche de contour	115
VI.4.3. Organisation général de la transformation Tr(3)	116
VI.4.4. Transformation Tr(4)	118
VI.5. Conclusions	119

CHAPITRE VII.

- *PERFORMANCES DU SYSTEME ET PERSPECTIVES D'AMELIORATION*

VII.1. Performances générales du prototype	120
VII.1.1. Performances de rapidité	120
VII.1.2. Limites de sensibilité	122
VII.1.3. Reproductibilité des résultats	125
VII.2. Perspectives d'améliorations et de généralisation	130
VII.2.1. Acquisition de formes quelconques	130
VII.2.2. Amélioration de la recherche de contour	
VII.3. Conclusions	138
 Conclusion générale	 139
Bibliographie	141

