NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Compression d'image ultrason par codage adapté au type d'information

par
Jean-François Thivierge

Une thèse
présentée à
l'Institut Ottawa-Carleton
de Génie Electrique
en complément du programme M.Sc.A. en
Génie Electrique

ISBN 0-315-75023-5

Je déclare être le seul auteur de cette thèse.

J'autorise l'université d'Ottawa à prêter cette thèse à d'autres institutions ou individus pour des fins de recherche.

Jean-François Thivierge

J'autorise également l'université d'Ottawa à reproduire la thèse par photocopie ou par d'autres moyens, en tout ou en partie, à la demande d'autres institutions ou individus pour des fins de recherche.

Jean-François Thivierge

L'université d'Ottawa requière la signature de toute personne utilisant ou photocopiant cette thèse. S'il vous plait signer ci-dessous en indiquant votre adresse et la date.

Résumé

Nous présentons une méthode de compression des images ultrasons sans perte significative d'information. Cette méthode s'inspire des méthodes de deuxième génération. Les différents groupes d'information sont tout d'abord extraits de l'image à l'aide d'un algorithme de segmentation. Trois groupes d'information sont identifiés soit l'échelle de gris, l'image ultrason et l'image de fond.

Chaque groupe d'information est alors traité individuellement, afin d'exploiter les caractéristiques de chacun d'eux pour leur compression. La notion d'information est redéfinie pour chacun d'eux afin d'optimiser la compression sans perte d'information.

Nous obtenons avec cette méthode un ratio de compression de 4 : 1, et ce, sans perte significative d'information. Nos images initialement codées sur 8 bits/pixel peuvent être codées en moyenne sur 1.952050 bits/pixel.

A mes parents

Remerciements

Je voudrais exprimer mes remerciements à mon directeur de thèse, Dr Morris Goldberg pour ses précieux conseils et son encouragement.

Je voudrais remercier aussi le département d'ultrason de l'hôpital Civic d'Ottawa, et plus particulièrement Mme Peggy Reed pour leur indispensable collaboration.

De plus, je voudrais remercier Northern Telecom pour leur support financier qui a rendu cette thèse possible.

L'aide et le support de mes collègues et amis et des membres du personnel du département de génie électrique furent grandement appréciés.

Finallement, je voudrais remercier mes parents pour la révision précieuse de ce texte ainsi que le support et l'encouragement qu'ils m'ont manifestés tout au long de ces études.

Sommaire

Liste des Figures

Liste des Tableaux

Chapitre 1

Introduction

De nos jours, les hôpitaux produisent un nombre croissant d'information médicale sous formes diverses. Il devient donc de plus en plus difficile de manipuler cette information. Une façon de faciliter l'accessibilité à cette information et de permettre une certaine flexibilité de son traitement est de numériser l'information pour la conserver dans un système informatisé. L'espace mémoire requis pour conserver toutes ces informations est considérable. Les images médicales demandent spécialement beaucoup d'espace mémoire. Plusieurs techniques de création d'images sont utilisées dans les hôpitaux. L'une d'elle est la création d'image par ultrason et c'est celle qui retiendra notre attention dans cette étude.

Il devient donc nécessaire de développer des techniques pour réduire cet espace-mémoire. Plusieurs méthodes de compression des images ont été développées. Les méthodes de première génération exploitent la forte redondance du signal et l'insensibilité relative de l'oeil à certaines distortions. Les exigences du milieu médical ne permettent cependant aucune perte d'information. L'introduction de toute distortion nous est donc interdite. De plus la redondance du signal ne doit être utilisée que de façon à n'engendrer aucune perte d'information. Récemment une seconde génération de méthodes de codage a vu le jour. Celles-ci considèrent l'image non comme un groupe de pixels, mais plutôt comme un ensemble de régions relativement homogènes, qui souvent correspondent aux différents objets présents dans l'image. Une segmentation de l'image est effectuée afin de détecter les différentes régions et des méthodes de codage adaptées à chacune de ces régions sont utilisées. Dans notre approche du problème,

nous retenons les éléments principaux de cette approche en les adaptant aux caractéristiques propres des images ultrasons.

Le chapitre 2 passe en revue les approches principales de segmentation des images, soit la classification des pixels, la détection des contours, et la formation des régions. Le chapitre 3 passe en revue les méthodes de codage de première et seconde génération et les différents critères utilisés pour l'évaluation des performances de compression. Le chapitre 4 présente dans un premier temps une brève description des principes physiques reliés à la formation des images ultrasons, et la technologie impliquée. Les caractéristiques des images ultrasons ainsi que la stratégie générale de compression que nous allons utiliser sont ensuite présentées. Nous retrouvons au chapitre 5 l'algorithme de segmentation que nous avons développé ainsi que les résultats obtenus avec nos quatre images tests. Au chapitre 6, pour chaque région de l'image, nous présentons les différentes techniques de compression que nous avons évaluées suivies de l'analyse des résultats obtenus. Au chapitre 7 nous présentons les conclusions de notre étude et quelques suggestions.

Chapitre 2

Méthodes de segmentation

La segmentation est une opération complexe. Traiter l'information brute est une chose, en extraire une certaine compréhension en est une autre. Nous pouvons diviser l'opération de la segmentation en deux tâches distinctes soit la segmentation proprement dite et l'interprétation. Riseman et Arbib [25] y réfèrent sous les noms de "traitement de bas niveau" et de "traitement de haut niveau". L'interprétation demande la considération de plusieurs facteurs qui, au niveau de l'information brute, demanderait un nombre considérable d'opérations. Nous préférons donc restreindre cette tâche à une information prétraitée. Ce prétraitement a pour but d'extraire de l'information brute, une information de deuxième niveau, soit une information plus structurée. Celle-ci peut consister en des régions, des limites, ou des contours. Des lois syntaxiques établies, à partir des connaissances que nous avons sur nos images et sur le genre d'information (régions, limites, contours) extraites par le prétraitement de l'information brute, sont utilisées afin d'interpréter cette information de deuxième niveau.

Nous avons donc deux étapes soit l'extraction d'une information structurée à partir de l'information brute et le traitement syntaxique de cette première. Dans la première étape nous organisons l'information sous une forme structurée pour ensuite l'interpréter dans la deuxième étape.

Dans l'étape de l'extraction d'une information structurée, la seule valeur de l'intensité du pixel ne suffit pas toujours à relier entre eux les pixels d'une même région. Nous devons

dans certains cas extraire une propriété qui est fonction d'un ensemble de points voisins. Il s'agit de la texture.

2.1 Techniques de Segmentation

Les techniques de segmentation ont pour but d'extraire une information structurée. Il est possible de les classer selon l'approche qu'elles utilisent en trois classes : les techniques de classification de pixel, les techniques de détection de contour, et les techniques de formation de région.

2.1.1 Techniques de classification de pixel

Haralick and Shapiro [10] regroupèrent ces techniques sous le nom de "measurement space guided spatial clustering techniques" (techniques par extraction de "clusters" dans un espace de mesure) ce qui illustre mieux en quoi elles consistent. L'approche utilisée est de classer chaque pixel individuellement, uniquement en fonction de la valeur qui lui est associée dans un espace de mesure. Ces techniques présument que les régions se manifestent sous forme de "clusters" dans l'espace de mesure. Un schéma bloc des étapes des techniques de classification est présenté à la figure 2.1. La première étape consiste à créer un espace de mesure qui

Figure 2.1: Technique de classification

est habituellement un histogramme. Ensuite, nous extrayons les "clusters", pour finalement effectuer la correspondance entre les pixels et les niveaux de gris constituant chacun des "clusters". Les méthodes diffèrent par la façon par laquelle l'espace de mesure est formé et les "clusters" extraits.

Méthodes de construction des histogrammes

L'espace de mesure le plus simple est l'histogramme des niveaux de gris des pixels de l'image. Celui-ci consiste en un recensement du nombre d'occurrences, à travers l'image, de chaque niveau de gris. Nous noterons cette fonction par "h(i) où "i" correspond au niveau de gris et "h(i)" au nombre d'occurences pour cette intensité. Il n'est cependant pas toujours facile de distinguer les différents "clusters" dans un tel histogramme. Diverses causes, telles que des limites entre régions bruitées, ou un éclairage non-homogène des objets lors de la création de l'image peuvent être à l'origine d'une telle situation. Afin d'éliminer ce problème, des méthodes de création d'histogrammes combinant d'autres informations au niveau d'intensité des pixels ont été développées. Chow et Kaneko [4] proposent une technique qui consiste à créer un histogramme différent pour chaque pixel à partir des pixels de son voisinage et d'extraire un seuil pour chacun d'eux. Pour éviter qu'un nombre considérable de calculs ne soient nécessaires, des histogrammes ne sont créés que pour un échantillonnage de points pour lesquels un seuil est déterminé, alors que les seuils pour les autres points sont calculés par interpolation.

En émettant l'hypothèse que les pixels constituant les frontières entre régions sont d'intensité intermédiaire, l'élimination de ces points de l'histogramme devrait contribuer à une meilleure séparation des "clusters". Weska et Rosenfeld [36] proposent de créer l'histogramme qu'à partir des pixels de faible gradient ou de forte valeur de Laplacien. L'hypothèse ici est que les pixels de fort gradient ou de faible valeur de Laplacien sont fortement susceptibles de constituer les points frontières. Panda et Rosenfeld [23] ont suggéré de créer deux histogrammes soit un à partir des pixels de faible gradient et un autre à l'aide des pixels de fort gradient. L'histogramme des pixels de faible gradient devrait présenter des vallées bien définies entre les "clusters" alors que l'histogramme des pixels de fort gradient devrait présenter un pic à ces mêmes positions. L'utilisation de ces deux histogrammes devrait renforcer la décision quant au choix d'un seuil.

Méthodes d'extraction des "clusters"

La méthode la plus simple d'extraction des "clusters" consiste à détecter les vallées les séparant. De bons résultats sont à espérer si l'histogramme ne présente pas de variation brusque et si la division entre les clusters est bien marquée. La plupart du temps ce n'est pas le cas et, cette façon de faire, extrait un nombre trop élevé de "clusters". Une façon de prévenir ce problème est d'évaluer l'importance relative des vallées. Ceci peut être fait en évaluant la profondeur des vallées. La profondeur d'une vallée peut être définie comme étant la différence entre le fond de la vallée et le plus faible de ses deux sommets adjacents [10]. Si nous savons combien de "clusters" nous devons obtenir, disons "n", nous pouvons considérer que les "n" vallées les plus importantes. Un lissage de l'histogramme peut également être effectué afin d'éliminer les vallées de moindre importance.

Wilson and Span [37] ont proposé une méthode différente pour extraire les "clusters". Ils la nomment "Local Centroid Clustering". L'idée est de modifier l'histogramme en déplaçant les valeurs des probabilités à la position de leur "centroïde local" itérativement jusqu'à ce qu'aucun changement dans l'histogramme ne soit observé. Le "centroid local" en chaque position x de l'histogramme est défini par :

$$\mu(x) = x + \frac{\int_{-W}^{W} X p(x+X) dX}{\int_{-W}^{W} p(x+X) dX} \tag{2.1}$$

ce qui n'est rien d'autre que le centre de masse de l'histogramme calculé sur une fenêtre de dimension 2W + 1. Une "table de mise en correspondance" est créée afin de noter les différents déplacements. Un "cluster" est assigné à chaque centre de masse existant à la fin du processus. Toutes les intensités des probabilités y ayant contribuées constituent celles du cluster. La qualité du résultat avec cette technique est cependant fortement dépendante du choix de la dimension de la fenêtre.

Ohlander [21] a proposé un algorithme d'extraction des "clusters", récursif. Celui-ci consiste à extraire les "clusters" des histogrammes contruits à partir de portions de l'image obtenues par différents masques. Au départ le premier masque est constitué par l'image

entière. Nous en extrayons les régions connexes appartenant au même "cluster" qui elles constitueront à leur tour les nouveaux masques. Ce processus est répété sur ces nouveaux masques jusqu'à ce qu'un seul "cluster" puisse être extrait de chacun des masques.

Les techniques de classification de pixel fonctionnent bien lorsque les différents objets ont un niveau de gris homogène et différent d'un objet à l'autre. Le problème principal vient de sa procédure de regroupement des pixels qui ne tient que très peu compte de l'aspect spatial. Ceci résulte en des frontières bruitées, en des régions fragmentées, ou en de petites régions isolées. L'avantage avec cette technique vient de ce même état de chose qui nous permet de relier des régions non connexes par le seul fait qu'elles originent du même cluster.

2.1.2 Techniques de Détection de Contours

Dans cette approche nous nous intéressons aux propriétés qui caractérisent les points frontières. En d'autres mots, nous cherchons à localiser les discontinuités. Les techniques de détection des contours utilisent différents opérateurs qui détectent ou mettent en évidence la variation de l'intensité aux limites des différentes régions. Deux approches sont possibles, soit la différenciation ou le "masquage adapté" (template matching).

Approche par Différenciation

Les contours sont caractérisés par un rapide changement de propriété caractérisant les pixels de chaque région. Ceci peut se traduire par une forte valeur de la dérivée première ou un passage par zéro de la dérivée seconde.

Un opérateur de différenciation est habituellement utilisé pour le cas des signaux numériques pour remplacer la dérivée. Nous présentons ici deux séries d'opérateurs qui peuvent être utilisés comme substituts de dérivée première.

$$\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \text{ et } \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \text{ et } \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

La première série effectue une différenciation dans la direction horizontale et verticale alors que la deuxième, connue sous le nom d'opérateurs de Roberts, effectue une différenciation selon une diagonale de 45 et 135 degrés. L'amplitude du gradient peut être évaluée en calculant la valeur RMS des deux valeurs de différenciation de direction orthogonale. Rosenfeld [27] nous a montré cependant que l'utilisation du maximum des valeurs absolues des différences nous donnait une réponse plus précise et demandait des calculs plus simples. Rosenfeld [27] et Davis [5] nous présentent un survol important de différents opérateurs de ce type.

Un des problèmes majeurs avec ces opérateurs est leur sensibilité au bruit. Le bruit se manifeste habituellement par un changement rapide d'intensité. C'est cette même caractéristique que l'on associe aux contours, de là une confusion possible entre le bruit et les contours. Afin de réduire les effets du bruit, nous pouvons lisser l'image avant l'utilisation des opérateurs, ou d'une manière équivalente utiliser un opérateur qui calcule une différence de moyennes locales. Le calcul de la moyenne locale doit être effectué sur des régions qui ne se recoupent pas, afin de ne pas réduire l'intensité du signal de différence. Malheureusement ces opérateurs produisent des contours flous. Afin de réduire cet effet, nous pouvons supprimer les points ne constituant pas des maximums locaux dans la direction perpendiculaire au contour. Une autre façon de réduire l'effet flou est de ne calculer la moyenne que dans la direction du contour. Les opérateurs de Prewitt utilisent cette technique.

$$\frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ et } \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Des moyennes pondérées peuvent être également utilisées. Un exemple simple est l'opérateur

de Sobel.

$$\frac{1}{4}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ et } \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Mouga [19] présente différents filtres de "lissage" pondérés et leur version dérivative. Pour n'en nommer que quelques-uns, nous avons le filtre Shen, le filtre Moyenne, le filtre Gaussien et le filtre Deriche. Il nous présente également une méthodologie pour la création d'algorithme de détection de contours utilisant de tels filtres.

La dérivée seconde, pour sa part, est très souvent approximée par le Laplacien.

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

Le Laplacien est proportionnel à $f - \bar{f}$, où $\bar{f}$ consiste en la moyenne locale du signal f. Marr et Hildreth [17] utilisent le filtre gaussien avec l'opérateur Laplacien comme opérateur de dérivée seconde. L'opérateur résultant $\nabla^2 G$ est décrit par l'équation suivante.

$$\nabla^2 G(r) = -1/\pi\sigma^4[1 - r^2/2\sigma^2]\exp(-r^2/2\sigma^2) \tag{2.2}$$

Imme [12] présente une technique intéressante qui utilise l'effet de lissage du filtre gaussien mais d'une façon plus empirique. Il fait l'hypothèse qu'entre chaque maximum et minimum il existe un contour. Il commence par lisser l'image en la filtrant par un filtre gaussien. Puis une détection des maximums et minimums est effectuée sur l'image lissée. La position des contours est définie par la position du gradient maximal. Le calcul du gradient est effectué sur l'image originale afin de prévenir le décalage des maximums et minimums et afin d'extraire l'amplitude réelle du gradient. Cette méthode résoud quelques-uns des problèmes les plus couramment rencontrés avec les méthodes de détection de contours comme l'épaisseur des contours et les manques dans les tracés des contours.

Approche par Masquage adapté

L'idée ici est d'associer la forme d'un contour à un patron d'une série de patrons possibles. Chaque patron est représenté par un masque différent. Tous les masques sont appliqués à la position d'intérêt afin de déterminer lequel produit la plus forte réponse. La valeur de la plus forte réponse est comparée avec un seuil en dessous duquel nous n'associons aucun contour en cette position. En toute position où la valeur du seuil est dépassée un patron de contour est conservé. Ces opérations sont répétées en toutes les positions de l'image.

Différentes séries de patrons ont été développées. Maître [15] présente une liste qui comprend en autre la série de Kirsch, Babu, Morgenthaler, Hammel and Prewitt. Tous sont construits de manière à produire une réponse nulle pour une région uniforme et une réponse maximale si une correspondance parfaite est trouvée. Rosenfeld [27] note que les contours peuvent être modélisés par des différences de second ordre. Ceci revient à dire que plusieurs des opérateurs présentés dans la section précédente pourraient être utilisés comme base pour définir des séries de masques. Nous avons entre autre la série de l'opérateur de Prewitt, de Sobel, et de Kirsch.

2.1.3 Techniques de Formation des Régions

L'approche ici consiste à regrouper les pixels en régions selon les propriétés qui les relient ensemble. Ce regroupement est fait dans le domaine spatial de l'image. Les techniques de formation des régions sont classées selon trois approches qui sont : le partage, la réunion, et le partage et réunion.

Approche par réunion

Dans l'approche par réunion nous commençons avec une image initialement divisée en pixels ou en petites régions fortement uniformes. Une mesure de la validité du regroupement des régions adjacentes, basée sur l'homogénéité de la région résultante, est définie. Les paires de

régions adjacentes rencontrant le mieux ce critère d'homogénéité sont réunies. Les techniques de réunion diffèrent d'une à l'autre par la façon dont les régions sont comparées et réunies, et par le critère d'homogénéité utilisé. Plusieurs critères mesurant l'homogénéité ont été développés. Nous les examinerons plus tard. Les régions sont créées selon un mode séquentiel ou omnidirectionnel. Dans le mode séquentiel, l'image est balayée ligne par ligne et chaque pixel ou petite région est réuni ou non avec une région adjacente existante. Suk et Chung [31] proposent une technique de segmentation selon un mode séquentiel. Dans cette méthode tous les pixels sont examinés un à un et comparés avec leur trois voisins sur une fenêtre carrée de 2 x 2. Nous regroupons ces quatre pixels de façon à former le plus gros groupe de pixels à l'intérieur de cette fenêtre rencontrant un critère d'homogénéité. Nous privilégierons donc un regroupement de (4) pixels, puis (3,1) pixels, (2,2), (2,1,1) et finalement (1,1,1,1). Le critère d'homogénéité choisi est la différence maximale d'intensité des pixels regroupés, qui doit être inférieure à un seuil fixé. Ces trois pixels voisins étant déjà assignés à des régions, il est possible que le regroupement proposé demande la réunion de deux régions. Un autre test, basé sur la différence entre les intensités moyennes de chaque région, est effectué pour évaluer la validité de cette action.

Dans le mode omnidirectionnel la croissance des régions se fait plus ou moins d'une manière parallèle. Plusieurs pixels de départ sont choisis et la croissance est effectuée dans toutes les directions.

Approche de partage

Dans l'approche de partage l'image entière constitue notre région de départ. L'approche consiste à la diviser en régions homogènes. L'homogénéité de la région de départ est tout d'abord évaluée et une décision est prise quant à la nécessité d'une subdivision. Si une subdivision est nécessaire le même test d'homogénéité est effectué sur chacune de ces nouvelles régions et les subdivisions nécessaires sont effectuées itérativement jusqu'à ce que toutes les régions résultantes soient homogènes.

Il est peu réaliste de vouloir évaluer toutes les façons possibles de subdiviser l'image. Alors seulement certaines subdivisions sont considérées. Rosenfeld [27] nous donne deux exemples soit la bisection et la triangulation. La bisection consiste à diviser la région en quadrants alors que la triangulation consiste à la diviser en secteurs triangulaires dont les sommets se rencontrent en un point intérieur (voir figure 2.2).

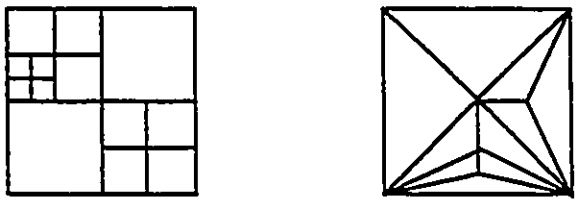

Figure 2.2: Méthodes de subdivision

Approche de partage et réunion

En effectuant seulement des opérations de partage, il est fort possible que le résultat final de la segmentation comprenne des régions adjacentes de caractéristiques similaires et qui, par conséquent, ne devrait constituer qu'une seule région. Si les opérations de partage sont suivies par des opérations de réunion, cette situation peut être corrigée. La méthode proposée par Horowitz et Pavlidis constitue un bon exemple. Ils ont défini trois opérations différentes. Premièrement, une étape de partition qui consiste à diviser également chaque région qui ne rencontre pas le critère d'homogénéité. Deuxièment, une étape de fusion qui consiste à réunir deux régions de même grandeur qui une fois réunies forment une région plus homogène. Et finalement, une étape de regroupement qui consiste à réunir des régions de toutes grandeurs qui une fois réunies rencontreront un critère d'homogénéité.

Approche pyramidale

Il existe une quatrième approche appelée pyramidale qui consiste à effectuer une segmentation sur différents niveaux de résolution. Nous effectuons une première segmentation sur l'image à basse résolution. L'information extraite à ce niveau est ensuite utilisée aux niveaux de résolution supérieurs et ainsi de suite jusqu'au niveau de pleine résolution. L'approche pyramidale est souvent considérée comme étant une technique en elle-même. Cependant, comme l'a noté Maitre [15], il s'agit plutôt d'un outil d'accélération qui n'introduit qu'une dégradation négligeable sur le résultat final. En effet, étant donné que cette approche est toujours combinée à une méthode de détection de contours ou de formation de régions, il est raisonnable de la considérer plutôt comme un outil accélérant. Plusieurs auteurs ont présenté des techniques basées sur l'approche pyramidale [8], [28], [29].

Critères d'homogénéité

Que nous choisissions l'approche de partage ou l'approche de réunion, la décision pour la formation des régions repose toujours sur un critère d'homogénéité. Nous présentons donc une brève liste de critères susceptibles d'être utilisés au niveau du pixel ou au niveau de la région.

- Critères au niveau du pixel
    1. Différence maximale de niveau de gris
    2. Différence maximale normalisée de niveau de gris
    3. Moyenne locale des niveaux de gris
    4. Variance locale des niveaux de gris
    5. Contraste
- Critères au niveau de la région

1. Intensité moyenne dans la région
2. Variance moyenne dans la région
3. Contraste dans la région
4. Dimension de la région
5. Forme de la région
6. Complexité de la forme de la region

le contraste peut être défini ainsi: contraste = (max - min)/(max + min).

2.2 Traitements Syntaxiques

Les traitements syntaxiques ont pour but d'organiser l'information obtenue lors de l'étape de segmentation. Ils utilisent les connaissances que nous avons sur les caractéristiques des contours et des régions telles la forme, l'orientation, etc.

2.2.1 Traitements Syntaxiques des contours

Les contours constituent les limites des régions. La première propriété qu'un contour doit avoir consiste en sa fermeture. Malheureusement il est plutôt rare que l'étape de segmentation produise de tels contours.

Des algorithmes de fermeture de contour ont été développés pour résoudre ce problème. Ceux-ci éliminent les points non désirables et ajoutent de nouveaux points pour combler les manques. Pour ce faire nous analysons les pixels susceptibles d'être ajoutés au contour. L'amplitude et la direction du gradient sont des caractéristiques qui sont utilisées lors de l'analyse des pixels voisins. Des caractéristiques plus globales telles la longueur des segments, la valeur moyenne des gradients, peuvent également être utilisées. Gonzalez et Wintz [7] ont proposé une analyse locale utilisant comme critères un seuil minimal de l'amplitude du

gradient et une différence maximale de la direction du gradient. Des procédures itératives, telle la "relaxation" section 10.5 dans [27], donnent quelquefois des résultats globaux de meilleure qualité.

Différentes méthodes de recherche dans des graphes évaluant les différents contours possibles tout en cherchant à minimiser une fonction du coût global ont été évaluées. Maître [15] nous présente une liste de telles méthodes. Les coûts informatiques (espace mémoire, temps d'exécution) pour ces méthodes sont souvent prohibitifs.

La transformée de Hough constitue une analyse globale intéressante considérant le regroupement des pixels en déterminant si ils occupent le tracé d'une forme particulière de courbe. Utilisé premièrement pour les lignes droites, cette méthode fut généralisée pour différentes formes de courbe. Le principe consiste à calculer un des paramètre définissant la courbe recherchée en fonction des autres paramètres en chaque point extrait par les opérateurs d'extraction de contours. Une matrice tient le compte de l'occurrence des combinaisons des différents paramètres définissant la courbe. La combinaison s'étant produite le plus souvent est retenue comme étant celle représentant le mieux la courbe définie par l'ensemble des points considérés. Pour le cas d'une ligne droite $y = mx + b$, nous pourrions choisir de calculer "m" en fonction de "b" en tous les points (x,y) extraits. Habituellement la représentation normale d'une droite est utilisée pour le calcul de la transformée de Hough.

$$\rho = x \cos\theta + y \sin\theta \tag{2.3}$$

Cette représentation est présentée à la figure 2.3.

Pour tous les points la valeur de ρ est calculée pour des angles succeptibles d'être ceux de la droite recherchée. Les valeurs de θ doivent être limitées à des valeurs discrètes afin de limiter la dimension de la matrice requise. La précision de la valeur calculée de rho doit également être limitée afin de limiter la dimension de la matrice. L'espace des paramètres θ et ρ est donc divisé en cellules de dimension correspondant à la valeur de l'incrément de l'angle et de la précision sur ρ. Un exemple d'une telle matrice est présentée à la figure 2.3. Les cellules sont tout d'abord initialisées à zéro. Pour chaque valeur de ρ calculée , la valeur

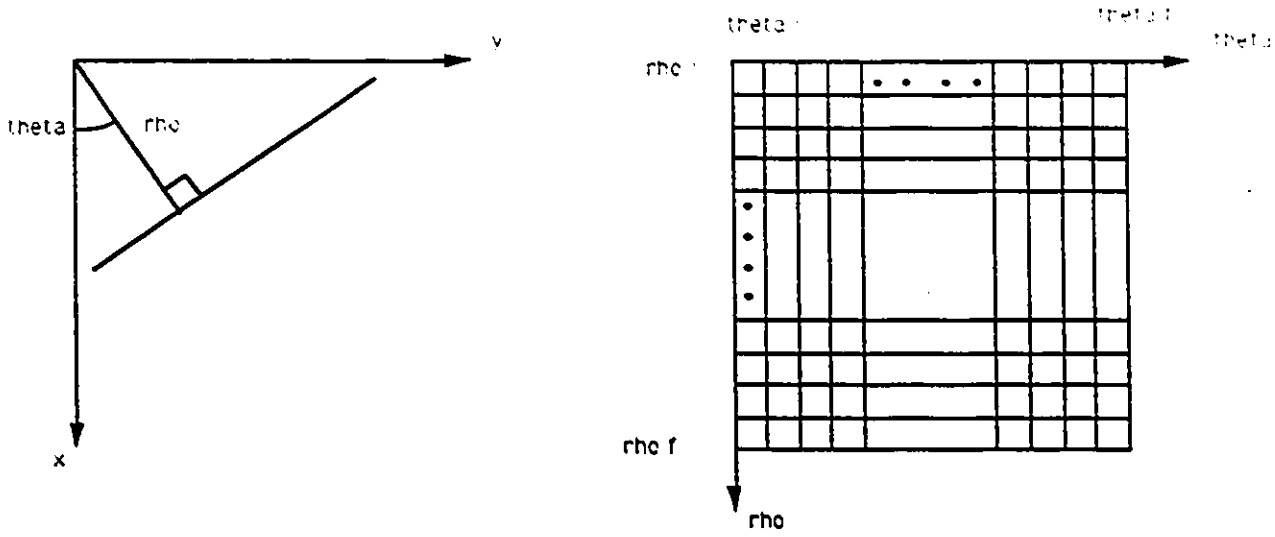

Figure 2.3: Représentation normale d'une droite, et matrice des cellules

de la cellule correspondant à la combinaison de ρ et de θ est incrémentée.

2.2.2 Traitements Syntaxiques des régions

Les images résultant de l'étape de segmentation comportent souvent de petites régions parasites, des régions avec des contours bruités, des régions ayant des trous, et des problèmes de sursegmentation qui consiste en une trop grande division des régions.

Pour les images binaires, l'érosion et la dilation peuvent être utilisées pour adoucir les contours des régions et pour éliminer les trous. L'érosion consiste à inverser l'intensité des pixels blancs ayant un voisin d'intensité noir. La dilation est le procédé inverse, soit d'inverser l'intensité des pixels noirs ayant un pixel blanc comme voisin. Des opérations successives d'érosion et de dilation adoucissent la forme des régions.

Lorsque nous connaissons les propriétés morphologiques des régions à extraire, les opérateurs de morphologie mathématique peuvent être utilisés. Bloch et Préteux [1] nous présentent une approche utilisant la morphologie mathématique pour l'extraction des lésions des images cérébrales par résonance magnétique. Le principe consiste à déterminer un

élément structural qui tient compte des propriétés des régions telles la forme, la grandeur et l'interaction entre les régions. Finalement, il est possible d'éliminer le problème de surségmentation en fixant une grandeur minimale des régions et en fusionnant les trop petites à leur voisines les plus similaires.

2.3 Critères d'évaluation de la performance de la segmentation

Il est important de pouvoir évaluer les résultats obtenus à l'aide des différents algorithmes de segmentation. Il est donc nécessaire d'établir des critères sur lesquels nous pouvons juger les résultats.

2.3.1 Critères pour les méthodes d'extraction de contours

Un survol intéressant de méthodes d'évaluation des performances des algorithmes de détection de contours est présenté par van der Heyden [34]. On y retrouve les critères utilisés par différents auteurs. Entre autre Abdou et Pratt ont basé leur critères sur les trois points suivants:

1. Contours non-détectés.
2. Bruits classés comme contour.
3. Erreurs de localisation.

Kitchen and Rosenfeld ont pour leur part suggéré d'utiliser une mesure de la cohérence locale des contours. Ceci consiste en une combinaison de mesure de la minceur et de la continuité des lignes de contours. Finalement, van der Heyden nous propose une liste d'aspects à partir desquels un ensemble de critères devraient être construits. Ces aspects sont :

1. La probabilité de faux contours et de contours omis.

2. La concentration de faux contours et de contours omis.

3. La moyenne et la variance des erreurs de positionnement des contours.

4. L'épaisseur des contours.

Il suggère également de pondérer chacun de ces aspects selon l'application.

Un des ensembles de critères largement cité dans la littérature est celui défini par Canny [2]. Il consiste en les trois critères suivants :

1. Bonne détection (contours non-détectés ou faussement détectés).

2. Bonne localisation (déplacement par rapport à la position idéale).

3. Réponse unique (largeur des contours).

2.3.2 Critères pour les méthodes de formation de région

Lorsque nous utilisons des algorithmes de formation de région, des critères plus appropriés peuvent être utilisés. Shen and Rosenfeld [29] utilisent un critère qui se définit comme le pourcentage de pixels classés dans une région n'appartenant pas à cette région et le pourcentage de pixels manquant dans cette région.

2.4 Techniques d'extraction de texture

La texture peut être définie comme une propriété locale évaluée sur un voisinage de pixels. "The major problem for all segmentation techniques is texture" (Le problème majeur de toute technique de segmentation est la texture) [25]. Ceci est du au fait que les textures ne sont pas définies au niveau du pixel mais plutôt sur un voisinage de pixels. Pour les régions texturées, contrairement aux régions homogènes où le niveau de gris est suffisamment stable pour constituer une propriété commune à tous les pixels d'une région, nous devons trouver

une propriété sur un voisinage de pixels qui sera commune à tous les pixels de la même région.

La grandeur du voisinage choisi peut devenir un problème sérieux. Etant donné que nous ne connaissons pas a priori les dimensions du "motif", le choix d'un trop petit voisinage peut causer la méprise d'une structure interne pour une nouvelle région. Le voisinage doit donc être suffisamment grand pour éviter ce problème en restant le plus petit possible afin de réduire le temps de calcul. Un autre problème est celui du "fenêtrage" qui se produit lorsque la fenêtre délimitant le voisinage chevauche deux régions différentes. Ceci produit une valeur intermédiaire créant ainsi artificiellement une nouvelle région.

Il est important d'identifier les résultats désirés et de déterminer une méthode pour évaluer les résultats obtenus. Comme nous l'avons mentionné précédemment, nous désirons extraire une propriété qui sera homogène dans une même région. De plus nous voulons que cette propriété nous permette de dissocier une région d'une autre. Le calcul de la variance de la valeur de la propriété à l'intérieur d'une région connue, nous permettrait d'évaluer l'homogénéité. Cependant ceci ne tient aucunement compte de la capacité de dissociation des régions. Le critère de Fisher [6] tiend compte de ces deux aspects. Il se définit comme suit :

$$f = (\eta_1 - \eta_2)^2/(\sigma_1^2 + \sigma_2^2) \tag{2.4}$$

où η_i est la moyenne des intensités des pixels de la région "i" et σ_i est la variance. Nous devons donc établir une zone représentative pour chaque région. Ce critère donne une mesure de la distance entre les distributions des valeurs des propriétés des deux régions.

La mesure de texture peut être utilisée de deux façons différentes. Premièrement, elle peut être utilisée après segmentation afin d'identifier les régions extraites en comparant les valeurs de textures obtenues avec une liste de valeurs espérées. Une deuxième façon de l'utiliser est comme prétraitement avant la segmentation, afin de transformer l'image dans un domaine où la segmentation serait plus facile. C'est cette deuxième application qui retiendra notre attention. Rosenfeld et Kak [27] ont défini une procédure générale de segmentation utilisant

l'extraction de texture qui consiste à calculer une propriété locale en chaque point, lisser l'image résultante en effectuant une moyenne locale, pour finalement segmenter l'image.

Gonzalez and Wintz [7] ont classé les techniques d'extraction de texture selon trois approches : l'approche statistique, l'approche structurale, et l'approche spectrale. Nous allons utiliser cette classification dans notre analyse à laquelle nous ajouterons une quatrième approche soit l'approche par transformation.

2.4.1 Approche Statistique

L'approche statistique consiste à extraire une valeur statistique à partir du voisinage de la position évaluée. Les statistiques peuvent être basées sur le niveau de gris des pixels ou sur le niveau de gris des pixels occupant deux positions spécifiques par rapport à la position évaluée. La moyenne du niveau de gris, la variance du niveau de gris, les caractéristiques de la matrice de co-occurrence, le nombre d'extremum par unité de surface sont quelques-unes des propriétés que nous retrouvons avec cette approche.

Les deux premières propriétés ci-haut mentionnées sont relativement simples et ne demandent donc aucune explication. La matrice de co-occurrence est pour sa part créée en comptant le nombre d'occurrences de chaque combinaison possible de niveau de gris pour une paire de position relative à la position testée. Chaque ligne correspond à une des différentes intensités possibles dans la première position et chaque colonne correspond à une des différentes intensités possibles dans la deuxième position. L'élément $a_{i,j}$ contient donc le nombre d'occurrences de la combinaison de l'intensité "i" en position 1 et de l'intensité "j" en position 2. La dimension de la matrice est fixe et déterminée par le nombre d'intensités possibles. Cette matrice contient cependant beaucoup trop d'information. Pour en retirer quelque chose, nous devons calculer certaines de ses caractéristiques. Voici quelques-unes des caractéristiques les plus communes.

Probabilité maximale

$$max_{i,j}(a_{ij}) \tag{2.5}$$

Moment d'ordre k des différences d'élément

$$\sum_i \sum_j (i-j)^k a_{ij} \tag{2.6}$$

Moment inverse d'ordre k des différences d'élément

$$\sum_i \sum_j a_{ij}/(i-j)^k \quad i \neq j \tag{2.7}$$

Entropie

$$-\sum_i \sum_j a_{ij} log\ a_{ij} \tag{2.8}$$

Uniformité

$$\sum_i \sum_j a_{ij}^2 \tag{2.9}$$

La probabilité maximale nous donne la combinaison de niveau de gris la plus fréquente. Le moment d'ordre k des différences d'élément pour sa part sera faible si les valeurs les plus fortes de la matrice se situent près de la diagonale, ce qui signifie que les niveaux de gris des pixels ne sont pas très différents. Le moment inverse des différences d'élément lui nous donne une réponse inverse. L'entropie sera élevée si nous avons une distribution fortement aléatoire. Finalement, la valeur de l'indice d'uniformité sera faible lorsque tous les éléments de la matrice sont égaux, ce qui correspond à une uniformité parfaite. Toutes ces équations ont été tirées de [7].

La dernière propriété soit le nombre d'extremum par unité de surface peut être calculée en recensant le nombre d'extremum sur un voisinage de pixels. Les extremum peuvent être trouvés en détectant les changements de signe de la dérivée première se produisant simultanément dans la direction horizontale et verticale. Les extremum peuvent être détectés plus simplement en les définissant que sur une dimension soit par un balayage horizontal [9]. Un pixel à la position (i, j) est un minimum relatif si son niveau de gris g(i, j) satisfait :

$$g(i,j) \leq g(i,j+1) \text{ et } g(i,j) \leq g(i,j-1)$$

Un pixel à la position (i, j) est un maximum relatif si son niveau de gris g(i, j) satisfait :

$$g(i,j) \geq g(i,j+1) \text{ et } g(i,j) \geq g(i,j-1)$$

2.4.2 Approche Structurale

L'approche structurale consiste à extraire des valeurs basées sur la structure, la forme d'une primitive choisie. "A primitive is a connected set of resolution cells characterized by a list of attributes"(Une primitive est un ensemble de cellules caractérisées par une liste d'attributs) [9]. Pour le cas de techniques par morphologie mathématique la primitive sera un élément structural alors qu'elle sera un bloc de pixels adjacents de même intensité dans le cas de la technique "run length".

Matheron proposa [18] une méthode d'extraction de texture par morphologie mathématique. Dans celle-ci nous devons définir un élément structural qui consiste en une sélection de pixels dans une fenêtre. Celui-ci est translaté à travers l'image et en chaque position où les pixels de l'image correspondant aux pixels de l'élément structural ont "1" comme intensité , le pixel correspondant à cette position est fixé à "1". La mesure de la texture peut alors être obtenue en comptant le nombre de pixels fixés à "1" dans un voisinage. Comme on peut le comprendre cette technique est restreinte aux images binaires.

La mesure de texture par "run length" s'effectue en évaluant le nombre de pixels adjacents au pixel de la position évaluée ayant une même intensité et ce, selon une direction prédéterminée [9], ou selon deux directions orthogonales [9],[27].

2.4.3 Approche Spectrale

Cette approche utilise les caractéristiques des fréquences spatiales. Nous allons examiner deux différentes façons d'extraire une telle information soit la transformée de Fourier et la fonction d'autocorrélation.

La transformée de Fourier nous procure une information bidimensionnelle sur la répartition de l'énergie dans le spectre. Nous pouvons en extraire différentes caractéristiques tel le pic d'énergie prédominant. Cette information bidimensionnelle peut également être transformée en deux fonctions unidimensionnelles soit une représentant l'énergie en fonction de la distance radiale et une autre représentant l'énergie en fonction de la phase [7].

$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r)$$

$$S(\theta) = \sum_{r=1}^{R} S_r(\theta)$$

De ces deux fonctions nous pouvons calculer différentes propriétés telles la valeur maximale, la moyenne ou la variance.

La fonction d'autocorrélation est calculée en effectuant une convolution d'une portion de l'image sur elle-même. Si les primitives tonales de l'image sont relativement petites, la fonction décroitra plus rapidement que si elles sont relativement grandes. Le taux auquel la fonction décroit, ou le déplacement correspondant à une valeur spécifique de la fonction sont deux mesures qui peuvent être utilisées comme propriété [7],[27].

2.4.4 Approche par transformation

L'approche par transformation consiste à transformer l'image initiale dans un nouveau domaine. La mesure de l'activité , des transformations à base de dérivée, et différentes versions de filtre médian sont quelques exemples de propriétés utilisant cette approche.

L'activité peut être définie comme l'intensité de la fluctuation du signal. Celle-ci peut être mesurée en calculant la moyenne sur un voisinage de la réponse de différents opérateurs de différence comme le Sobel, l'opérateur "Robert cross" [9]. Une fonction plus complexe appelée "Busyness" a été présentée dans [27].

$Busyness = min(v_x, v_y)$, où

$$v_x = |A - B| + |B - C| + |D - E| + |E - F| + |G - H| + |H - I|$$

$$c_4 = |A - D| + |D - G| + |B - E| + |E - H| + |C - F| + |F - I| \quad (2.10)$$

(2.11)

où A, B, C, D, E, F, G, H, et I représentent les éléments de la matrice suivante :

$$\begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix}$$

Les transformations par dérivée sont également utilisées. Les opérateurs de dérivée première (Sobel, Robert cross) ou les opérateurs de dérivée seconde (Laplacien) peuvent extraire des valeurs qui après calcul de la moyenne locale seront suffisamment homogènes pour être utilisées comme mesure de texture [27].

Tout un ensemble de variantes du filtre médian peuvent être utilisées pour extraire la texture. Le filtre médian standard, le décompte du nombre de pixels d'intensité plus élevée, et le nombre de pixel d'intensité moins élevée que la moyenne, la valeur maximale et la valeur minimale des pixels, la valeur interquartile sont quelques-uns de ceux-ci [7],[27].

Chapitre 3

Techniques de compression de l'image

Les systèmes de compression d'information sont subdivisés en trois composantes: la modélisation des données, la modélisation des statistiques, et l'encodage [11].

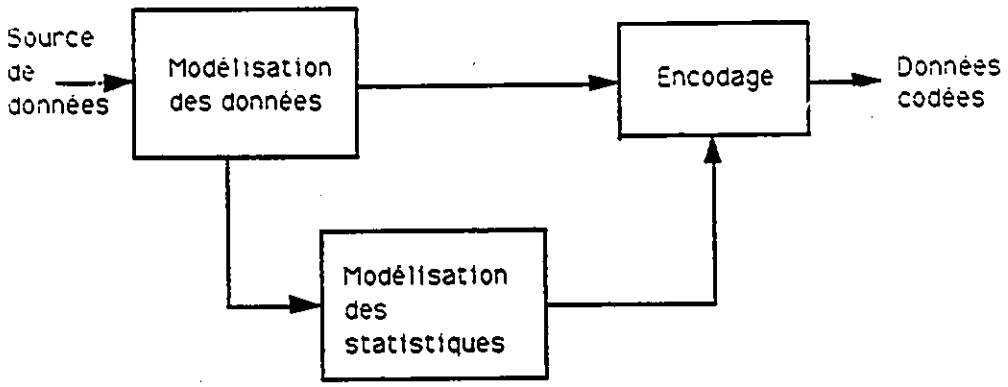

Figure 3.1: Système de compression d'information

La modélisation des données est la composante la plus importante car c'est à cette étape que l'on extrait l'infomation des données. Le résultat est un signal contenant l'information utile. La meilleure modélisation sera celle qui sera la mieux adaptée aux caractéristiques du signal. La plus simple modélisation consiste à prendre les données telles quelles comme signal à coder. Cependant cette modélisation n'atteint pas vraiment l'objectif de cette étape qui consiste en la réduction de l'information contenue dans le signal à coder à un minimum acceptable. Pour le cas des codages prédictifs, le signal résultant de la modélisation peut être

analysé à l'aide de la théorie de l'information qui nous donne une mesure de l'information présente dans chaque symbole qui se défini comme suit :

$$I = -\log_2 p_i \tag{3.1}$$

où "p" est la probabilité que ce symbole soit généré. L'utilisation du logarithme en base 2 nous donne une mesure en bit. On peut définir l'information comme une mesure de l'incertitude ou plus exactement de l'invraisemblance d'un état. Plus un état est invraisemblable, plus il contient de l'information (information = $\log(1/pi)$). Nous pouvons également la définir comme étant une mesure de l'effort à accomplir pour identifier cet état [22]. A titre d'exemple considérons N états différents tous équiprobables. L'information serait déterminée par le nombre d'étapes nécessaires, par la méthode la plus efficace, pour identifier de quel état il s'agit. L'information moyenne du signal, appelée entropie, est définie comme suit:

$$H = \sum -p_i \log_2 p_i \tag{3.2}$$

Il s'agit de la moyenne pondérée de l'information de chaque symbole. L'entropie du signal modélisé nous donne donc une mesure de l'efficacité de la modélisation. C'est ici qu'une certaine distortion peut être introduite lorsqu'une perte d'information est tolérable. Celle-ci est introduite de façon à minimiser son effet sur la qualité et à maximiser son effet sur la compression. Dans le cas des codeurs par transformation l'analyse du signal résultant dépend du nombre de coefficients sélectionnés et de la quantification effectuée sur ceux-ci. Nous verrons l'influence de ces paramètres sur l'efficacité de la compression plus loin.

La modélisation des probabilités n'est effectuée que dans le cas des codeurs prédictifs. Ceux-ci utilisent la connaissance de la distribution des probabilités des différents symboles à coder. Ne connaissant pas celle-ci, il est nécessaire de l'estimer. Trois types de modélisation des statistiques existent. Ils dépendent de la façon dont les probabilités sont estimées. Nous avons le type fixe, en deux étapes, et adaptif.

L'encodage consiste à assigner différents mots de code à tous les symboles possibles du signal résultant de la modélisation des données. Cette assignation peut dépendre des statistiques ou de leur importance relative, ou de la précision désirée sur chacun d'entre eux.

Les techniques de compression se divisent en deux grandes classes soit les techniques sans perte d'information et les techniques avec perte d'information. Les techniques sans perte d'information permettent de reconstruire exactement le signal de départ, alors que les techniques avec perte d'information introduisent une distortion jugée acceptable. Etant donnée que le type d'information que nous désirons compressée ne nous permet pas de perte d'information, nous restreindrons notre analyse aux techniques de compression sans perte. A l'intérieur de ces deux classes, les techniques sont classées selon que les données ont subies une transformation de domaine ou non. Les méthodes sans transformation s'applique sur les valeurs des pixels ou sur des valeurs directement dérivées de celles-ci, il s'agit du domaine spatial. Les méthodes avec transformation s'applique sur des coefficients extrait du signal résultant de la transformation. Comme nous l'avons vu précédemment, deux étapes sont toujours présentes dans tout système de compression soit la modélisation des données et l'encodage. La première génération de méthode de compression ont mis l'emphase sur l'encodage en considérant plutôt le pixel comme l'élément d'information. La seconde génération de méthodes de compression ont pour leur part mis l'effort sur la modélisation des données.

3.1 Première génération de méthodes de compression

3.1.1 Méthodes dans le domaine spatial

Les méthodes les plus simples utilisent la connaissance de la distribution des probabilités des différentes intensités présentes dans l'image. Elles accomplissent une compression en assignant de courts mots de code aux intensités les plus fréquentes et de plus long mot de code aux intensités les moins fréquentes. Le code de Huffman est optimal en ce sens.

Méthodes de modélisation des données

La modélisation la plus simple consiste à prendre telle quelle la valeur de l'intensité de chaque pixel comme symbole à coder. Cette modélisation n'est cependant pas très efficace.

Les méthode prédictives tentent de réduire davantage cette valeur d'entropie en prédisant l'intensité du prochain pixel et en ne codant que le signal de différence. Les différences d'un prédicteur à l'autre sont fixées par les valeurs du signal auxquelles ils s'appliquent, le nombre de coefficients, leurs valeurs et le fait que celles-ci soient fixes ou adaptatives. Les prédicteurs utilisent la redondance du signal qui se manifeste, pour les images, dans les deux dimensions et dans le temps. Les valeurs du signal utilisées pour la prédiction peuvent être les valeurs des pixels adjacents de la même image ou la valeur du même pixel de l'image précédente. Pour des images contenant des détails fins et présentant peu de mouvement, les prédictions utilisant l'image précédente donnent de meilleurs résultats. A mesure que le mouvement devient plus important les prédictions utilisant les pixels de la même image donnent de meilleurs résultats [20]. Afin d'obtenir que des valeurs discrètes comme signal de différence, il est nécessaire d'arrondir le signal de prédiction à l'entier le plus près.

La technique "runlength" est une autre méthode de modélisation des données. Bien qu'elle ne constitue pas une méthode prédictive, elle utilise également la forte redondance du signal. Celle-ci se base sur le fait que pour des signaux relativement stable et à forte redondance, l'intensité des pixels demeurent constante sur plusieurs pixels. Nous pouvons donc coder cette séquence de pixels en ne conservant que l'intensité et le nombre de pixels où celle-ci reste stable. Afin de fixer le nombre de symboles que produira notre modélisation, nous devons fixer la longueur maximale des séquences de pixels. Chaque symbole à coder sera donc une combinaison d'une intensité et d'une longueur.

Méthodes de modélisation des statistiques

Étant donné que les encodeurs utilisés pour coder le signal résultant se servent de la connaissance des probabilités de chacun des symboles, il est nécessaire d'approximer cette connaissance en modélisant ces statistiques. Comme nous l'avons vu, trois types de modélisation existent soit le modèle fixe, le modèle en deux étapes, et le modèle adaptatif.

Le modèle fixe comme son nom l'indique consiste en une modélisation fixe dans le temps. Celle-ci est établie en effectuant une moyenne sur un très grand nombre de données. Comme le note Helman et Langdon [11], ce modèle ne compresse pas aussi efficacement l'information car il représente une valeur moyenne qui n'est pas spécifiquement adaptée aux données à compresser. Cependant, étant donné que le modèle est fixe, il résulte en des algorithmes de codage/décodage plus simples.

Le modèle en deux étapes tire son nom du fait que les données sont manipulées deux fois. Premièrement les données sont examinées afin de créer un histogramme. Celui-ci tient place de statistique et c'est à partir de lui que le code sera créé. Les données sont examinées pour une deuxième fois lors de leur encodage. Cette modélisation des statistiques donne à première vue de meilleurs résultats étant donné que les statistiques sont complètement adaptées aux données à coder. Cependant, le code n'étant pas fixe, il est nécessaire de considérer l'espace mémoire requis pour transmettre ou entreposer celui-ci.

Le modèle adaptatif s'adapte au fur et à mesure que les données sont codées. Il débute avec des valeurs arbitraires pour être modifié par chaque nouveau symbole codé. Celui-ci ne nécessite pas d'espace mémoire pour la transmission de code étant donné que le codeur et le décodeur débutent avec les mêmes statistiques et qu'ils sont modifiées par les données transmises. Ce modèle donnera de meilleurs résultats que le modèle en deux étapes si les statistiques varient à l'intérieur d'un même fichier.

Méthodes d'encodage

Pour le type de signal résultant des méthodes de modélisation dans le domaine spatial, les méthodes d'encodage à mot de code de longueur variable sont très utilisées. Ces méthodes consistent à assigner de courts mots de code aux symboles les plus fréquents et de longs mots de code aux symboles les moins fréquents. Le code de Huffman [16], qui est probablement le plus connu, est optimum dans ce sens. Nous savons qu'un code uniquement décodable peut toujours être remplacé par un code "prefix-free" ayant les mêmes longueurs de mots de code. De plus nous pouvons prouver que pour un code binaire optimum, les mots de code pour l'état le moins probable et le deuxième état le moins probable ne diffèrent que par leur dernier bit. Finalement, assumons connaître le code optimum pour une certaine source. Considérons maintenant une source identique sauf pour ce qui est de ses deux états les moins probables; qui eux remplacent un état de la source originale et se divisent sa probabilité. Le code optimum pour cette nouvelle source sera le même que celui trouvé pour notre première source à l'exception des codes pour les deux états les moins probables qui eux seront définis par l'ajout d'un "0" et d'un "1" au code de l'état dont leur probabilité est issue [16]. Par récurrence nous pouvons remonter jusqu'au cas où nous n'avons que deux états pour lesquels la solution est triviale. C'est ainsi que le code de Huffman se définit et se détermine pour chaque source.

Redfern et Hines [24] nous présentent un code qu'ils appellent "Shift level code" (code de changement de niveau). Ce code s'applique seulement au signal de différence produit par les prédicteurs. Il s'agit d'un code dont les mots de code sont définis sur plusieurs niveaux. Le premier niveau d'un code à deux bits se définit comme suit :

Code	*Signal correspondant*
00	passer au niveau inférieur
01	0 (signal de différence)
10	+1 (signal de différence)
11	passer au niveau supérieur

Il permet de coder un signal de différence de 0 à +1. Si la valeur est inférieure à 0 on transmet le code 00 et on passe au niveau inférieur, et si la valeur est supérieure à +1 on transmet le code 11 et on passe au niveau supérieur. Le premier niveau inférieur se définit comme suit :

Code	*Signal correspondant*
00	passer au niveau inférieur
01	-3 (signal de différence)
10	-2 (signal de différence)
11	-1 (signal de différence)

et le premier niveau supérieur comme suit :

Code	*Signal correspondant*
00	+2 (signal de différence)
01	+3 (signal de différence)
10	+4 (signal de différence)
11	passer au niveau supérieur

Les codes obtenus à chaque niveau sont mis bout à bout afin de former le mot de code complet du signal de différence à coder. Pour une image codée sur 256 niveaux de gris nous avons besoin de 85 niveaux inférieurs et 85 niveaux supérieurs. Pour que ce code soit efficace, il est nécessaire que le signal de différence soit concentré dans les faibles valeurs. Il est possible de définir un code similaire sur quatre bits.

La prochaine méthode de codage utilise un concept similaire. Suzan dans [32] la nomme "differences technique" (code des différences communes). Dans celle-ci les quinze valeurs les plus probables sont codées sur 4 bits. Un mot de code est réservé pour représenter toutes les autres valeurs. Il sert de mot de code d'exception et est suivi de la valeur intégrale du pixel codée sur 8 bits. Le code est donc le suivant :

Code	*Signal correspondant*
0000	mot de code d'exception
0001	-7 (signal de différence)
0010	-6 (signal de différence)
0011	-5 (signal de différence)
0100	-4 (signal de différence)
0101	-3 (signal de différence)
0110	-2 (signal de différence)
0111	-1 (signal de différence)
1000	0 (signal de différence)
1001	+1 (signal de différence)
1010	+2 (signal de différence)
1011	+3 (signal de différence)
1100	+4 (signal de différence)
1101	+5 (signal de différence)
1110	+6 (signal de différence)
1111	+7 (signal de différence)

Le schéma général des méthodes de compression utilisant un prédicteur est présenté à la figure 6.1.

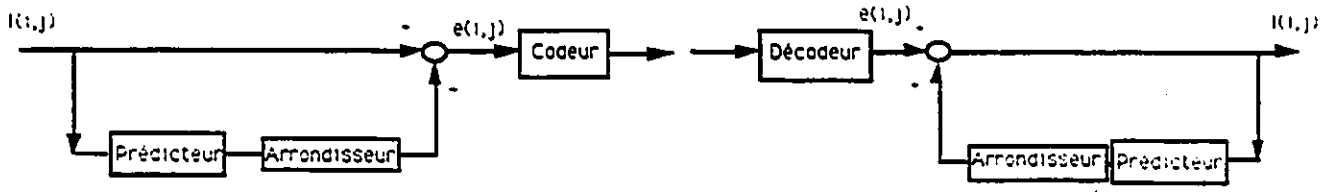

Figure 3.2: Schéma général de codage du signal de différence

3.1.2 Méthodes dans un domaine transformé

Comme nous l'avons mentionné précédemment, les méthodes de codage dans un domaine transformé s'effectue en deux étapes soit la transformation des données et le codage des coefficients extraits de ce nouveau domaine. En pratique les transformations ne sont pas appliquées sur l'image entière, à l'exception du "subband coding", afin de réduire le temps de calcul. L'image est plutôt divisée en petits blocs. Des grandeurs de blocs standard sont de l'ordre de 4x4, 8x8, ou 16x16 [26]. Plusieurs méthodes de transformations existent. La

transformée de Karhunen-Loeve est dite optimale, en ce sens qu'elle génère des coefficients qui sont complètement décorrélés [26]. Plusieurs autres transformations peuvent être utilisées telles la transformée du cosinus (DCT), la transformée de Fourier, la transformée de Hadamard [20].

Le problème avec les méthodes de compression par transformée comme le note Roos [26] est que lors du codage des coefficients ceux-ci doivent subirent une quantification qui elle introduit un erreur à la reconstruction du signal. Il est possible de réduire cette erreur en augmentant la quantification, cependant bien avant d'atteindre la reconstruction parfaite, l'efficacité aura été réduite considérablement. Le problème est résolu en transmettant également l'erreur sur le signal. Cette solution réduit quand même grandement l'efficacité de telles méthodes et les rend d'autant moins attrayantes.

3.2 Bref survol des méthodes de compression de seconde génération

Les deux principales étapes de toute méthode de codage sont la modélisation des données et l'encodage. Pour les méthodes de seconde génération l'effort est mis sur la modélisation plutôt que sur l'encodage. "Pixels are consequences of technical constraints in transforming an analog scene into digital data" (Les pixels sont une conséquences des contraintes techniques de la transformation du signal analogique en un signal numérique) [14]. L'utilisation d'entités plus près des vrais entités présentes dans l'image devrait conduire à une modélisation plus cohérente et à une meilleure compression.

Les contours et les textures sont un ensemble d'entités qui sont utilisés par différentes méthodes. Nous allons examiner deux méthodes utilisant ces entités soit le codage par décomposition directionnelle et le codage par segmentation. Kunt [13] présente une méthode par décomposition directionnelle. Nous décomposons l'image en images directionnelles, plus une image de basse fréquence. Les images directionnelles sont le résultat de filtres couvrant un secteur du domaine spatio-fréquentiel. Chaque secteur correspond à une direction. Les

images directionnelles sont extraites à l'aide des filtres directionnels et l'image de basse fréquence est extraite à l'aide d'un filtre passe-bas. Un filtre directionnel consiste en un filtre passe-haut enligné le long d'une direction principale. La réponse en fréquence idéale du ième filtre directionnel est donnée par :

$$H(f,g) = \begin{matrix} 1 & \text{si v(i)} \leq \tan(g/f) < v(i+1) \text{ et } (f^2+g^2) \leq \rho_c^2 \\ 0 & \text{autrement} \end{matrix}$$

avec $v(i) = (i-1)/2\pi$, $v(i+1) = (i+1)/2\pi$ et $\mid f \mid, \mid g \mid < 0.5$ où f et g sont les fréquences spatiales, et ρ_c la fréquence de coupure du filtre passe-bas. La superposition de toutes les images directionnelles et de l'image basse fréquence donne l'image de départ. Il nous reste à coder les images directionnelles et l'image basse fréquence. Les images directionnelles peuvent être représentées par la position et l'amplitude du gradient au position des passages par zéro. L'image basse fréquence contient des fréquences beaucoup plus faibles que l'image de départ. Nous pouvons donc la coder en la rééchantillonnant et en codant les pixels résultants.

Pour ce qui est des méthodes de codage par segmentation nous divisons d'abord l'image en régions. Chaque région est caractérisée par sa texture. Un prétraitement est nécessaire afin de réduire la granularité de l'image, prévenant ainsi une sursegmentation. L'image est alors codée en codant la position du contour et la texture de chaque région. Kunt [13] nous propose une méthode. Il utilise le filtre de l'inverse du gradient comme prétraitement. Il effectue la segmentation à l'aide d'une fusion-division adaptative. La texture est codée en approximant la forme générale du niveau de gris dans chaque région par une fonction polynomial à deux dimensions.

Ces deux méthodes sont des méthodes avec perte et par conséquent ne peuvent être utilisées dans notre application. Cependant l'idée générale peut être retenue et utilisée.

3.3 Critères d'évaluation des performances

Le critère habituellement utilisé est le ratio de compression. Celui-ci se défini comme suit :

$$ratio\ de\ compression = \frac{espace\ mémoire\ initial}{espace\ mémoire\ final} \tag{3.3}$$

Une autre façon d'évaluer la compression est en examinant la longueur moyenne par symbole soit :

$$\bar{n} = \sum_{i=1}^{N} p_i n_i \tag{3.4}$$

où p_i est la probabilité du symbole "i" et n_i est la longueur en bits du mot de code associé au symbole "i".

Redfern et Hines [24] définissent une mesure qui tient compte du temps nécessaire pour la compression, il s'agit du taux de compression.

$$taux\ de\ compression = \frac{ratio\ de\ compression}{temps\ de\ compression} \tag{3.5}$$

En effet, le temps de compression et de décompression s'avère souvent tout aussi important que le ratio de compression. Il est donc nécessaire d'évaluer un compromis acceptable entre ces deux facteurs.

Chapitre 4

L'ultrason et les images ultrasons

Différents types d'images, créées par différentes techniques, sont produites dans les centres hospitaliers. Pour ne nommer que les plus connues nous avons les rayons X, le CT scan, la résonnance magnétique et l'ultrason. Notre intérêt se limite ici aux images ultrasons. Les images ultrasons sont utilisées pour examiner les tissus mous comme les régions de l'abdomen et de la gorge, principalement en obstétrique et pour la détection du cancer. En 1990, 82 432 films d'images ultrasons ont été produits à l'hôpital Civique d'Ottawa. La loi et les exigences de la profession médicale exigent que les films soient conservés pour une période de cinq ans. Ceci revient à dire qu'après cinq ans d'opération un département d'ultrason aurait jusqu'à 412 160 films aux archives. De plus, chaque film contient 6 images, ce qui nous donne un total de 2 472 960 images. Il devient donc nécessaire de développer des moyens de coder et d'entreposer ces images efficacement.

L'hôpital Civique d'Ottawa utilise deux types d'équipement ultrason soit le "Ultramark 4" (UM4) et "Ultramark 8" (UM4) tous les deux construis par "Advanced Technology Laboratories" (ATL). Ils constituent un bon exemple de ce que l'on retrouve comme équipement dans un département d'ultrason.

L'appareil ultrason comporte un moniteur vidéo qui permet à l'opérateur de visualiser l'image en temps réel. Différents ajustements de l'appareil sont à la disposition de l'opérateur afin d'améliorer l'image visualisée. Un appareil photographique muni d'une interface permet

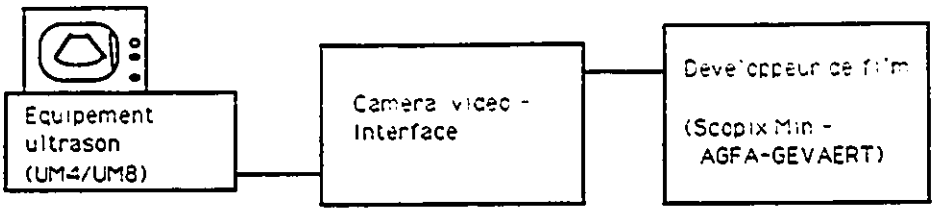

Figure 4.1: Organisation de l'équipement ultrason

de faire des copies de l'image sur film. Ce sont ces films qui constituent l'information que nous désirons coder et entreposer efficacement. L'organisation de cet équipement est présentée à la figure 4.1.

Il est tout d'abord nécessaire de connaître cette information soit les images ultrasons, afin d'en extraire des propriétés exploitables pour améliorer l'efficacité de leur entreposage.

4.1 Les Images Ultrasons

4.1.1 Principes physiques de la création des images ultrasons

L'imagerie par ultrason, comme son nom l'indique, utilise les ondes sonores pour créer des images de l'intérieur du corps humain. Examinons donc tout d'abord les caractéristiques de propagation des ondes sonores. Les caractéristiques de propagation dépendent du milieu dans lequel elles voyagent. Elles sont déterminées par l'impédance acoustique Z qui est définie par l'équation suivante :

$$Z = \rho V \tag{4.1}$$

où ρ est la densité du milieu de propagation et V est la vitesse de propagation du son dans le milieu. La vitesse de propagation V ainsi que l'impédance acoustique Z pour différents milieux, sont données au tableau 4.1.1.

Ces valeurs furent trouvées dans [33] et dans [30].

Comme pour la lumière, lorsque l'onde sonore passe d'un milieu à un autre et qu'il y

Milieu	*Vitesse de propagation* $V\ (m/s)$	*Impédance acoustique* $Z\ (gp/cm^2 \times 10^5)$
air	331	0.0004
eau	1500	1.48
foie	1570	1.65
os	4080	7.00

Table 4.1: Caractéristiques des milieux de propagation

a variation de l'impédance acoustique, une réflexion et une réfraction se produisent. Pour la réflexion, tout comme pour la lumière, l'angle de l'onde réfléchie égale l'angle de l'onde incidente. Pour la réfraction, l'angle de l'onde réfractée θ_t, peut être déterminé par l'équation suivante :

$$\frac{\sin\theta_i}{\sin\theta_t} = \frac{V_1}{V_2} \tag{4.2}$$

Les proportions de l'énergie de l'onde qui est réfléchie p_r, et qui est transmise p_t (réfractée) peuvent être calculées par les équations suivantes :

$$\frac{p_r}{p_i} = \frac{Z_2\cos\theta_i - Z_1\cos\theta_t}{Z_2\cos\theta_i + Z_1\cos\theta_t} \tag{4.3}$$

et

$$\frac{p_t}{p_i} = \frac{2Z_2\cos\theta_i}{Z_2\cos\theta_i + Z_1\cos\theta_t} \tag{4.4}$$

Le diagramme présenté à la figure 4.2 illustre bien ces phénomènes.

C'est le phénomène de réflexion qui est utilisé dans le processus de reconstruction de l'image. Un faisceau est tout d'abord émis par la sonde à travers le corps à examiner.

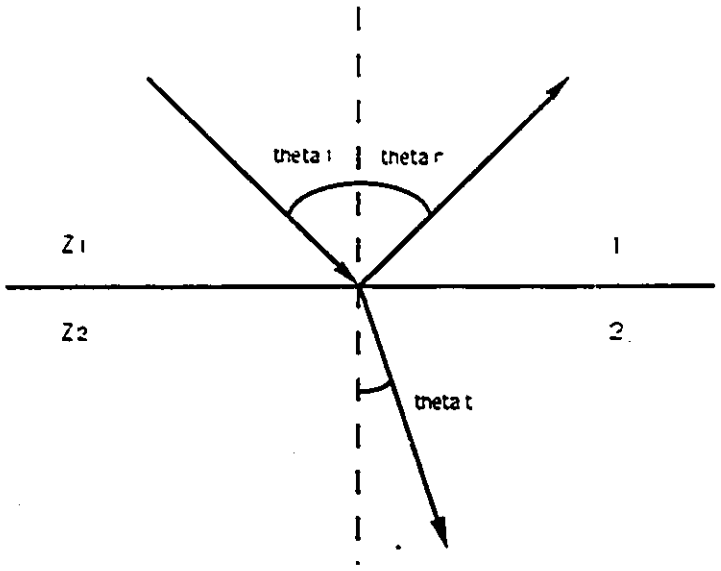

Figure 4.2: Phénomène de réflexion et de réfraction

Si l'onde sonore rencontre un changement d'impédance acoustique, une réflexion partielle se produit. Nous pouvons alors déduire la position où la réflexion s'est produite dans le corps, de la valeur du temps requis par l'onde pour parcourir l'aller et le retour. En assumant une vitesse moyenne de propagation de $1570m/s$, la position peut être calculée à l'aide de l'équation suivante : $y = \frac{t}{2} \times 1570m$ où t est le temps. En répétant cette opération en plusieurs positions le long d'une ligne droite, disons le long d'un axe "x", nous obtenons une coupe de l'image le long de cet axe.

L'onde est créée par un crystal contenu dans la sonde. C'est l'effet piézo-électrique qui est utilisé pour créer l'onde sonore. Il se décrit comme suit : l'application d'une différence de tension aux bornes d'un crystal produit une variation de sa largeur. Si la source de tension est alternative, le crystal oscillera et produira donc une onde sonore. Réciproquement, une variation de la largeur du crystal produit une différence de tension à ses bornes. C'est de cette façon que l'onde réfléchie est détectée. La sonde joue donc le double rôle d'émetteur et de récepteur alternativement. Dans les faits, une courte période d'émission est suivie d'une longue période de réception afin de permettre la réception de tous les échos, et ce en chaque position.

L'énergie de l'onde sonore diminue en fonction de la distance qu'elle parcourt. Le phénomène de réflexion où p_r/p_i de l'énergie est retournée vers la sonde n'est pas le seul effet qui contribue à cette réduction d'énergie. Un phénomène d'absorption par lequel une partie de l'énergie est convertie en chaleur, contribue également à cette réduction d'énergie. Cette réduction d'énergie occasionne un sérieux problème. En effet, une transition de milieu retournera une onde d'énergie proportionnelle à l'énergie de l'onde incidente et ce toujours dans le même rapport, cependant en région profonde l'énergie de l'onde incidente étant de faible intensité, l'onde retournée constituera un signal moins important que pour une transition équivalente en région peu profonde. Il est possible de compenser cet effet en appliquant un facteur d'amplification proportionnel à la distance. Un exemple d'une telle correction est présenté à la figure 4.3.

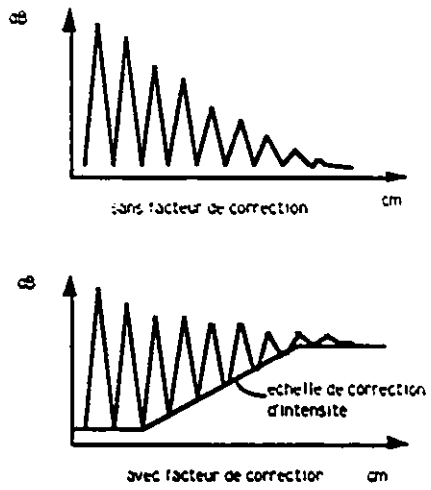

Figure 4.3: Correction de l'intensité du signal reçu

L'imagerie ultrason est utilisée en différent modes soit le mode A, TM, B et D. Dans le mode A, la sonde est fixe et le signal reçu indique les différentes transitions que rencontre le faisceau. Ce mode est utilisé pour effectuer des mesures selon une direction précise dans le corps. Dans le mode TM, l'information obtenue en mode A est étudiée en fonction du temps. Ce mode est utilisé pour détecter l'activité cardiaque et les mouvements du foetus. Dans le mode B, la sonde est déplacée manuellement le long d'une droite. L'information ainsi obtenue consiste en une coupe de l'intérieur du corps examiné. Finalement, en mode D le déplacement

est effectué automatiquement soit par un mouvement latéral ou une rotation de la sonde. Ce mode est le mode de création de toutes nos images tests. Plus d'information sur les images ultrasons et sur les principes de leur formation peut être trouvée dans [30] [33].

4.1.2 Caractéristiques des images ultrasons

L'image ultrason n'est pas la seule information présente sur les films produits par l'équipement ultrason. Tous les paramètres d'ajustement, ainsi que l'identification du patient, les résultats des différentes mesures effectuées y apparaissent sous forme alphanumérique. De plus une échelle de niveau de gris qui nous permet d'évaluer la qualité du développement du film est toujours présente. Egalement, une échelle correspondant à la correction d'intensité du signal en fonction de la distance d'origine de l'écho est présentée. Finalement, nous avons les pointeurs indiquant les positions des mesures effectuées. La disposition de ces différentes informations varie en fonction de l'équipement utilisé soit dans notre cas l'appareil UM4 ou UMS. Une image typique obtenue par chacun de ces deux appareils est présentée à la figure 4.4.

Notons finalement que la forme de l'image ultrason varie d'une image à l'autre. En fait la forme de l'image varie en fonction de la sonde utilisée.

4.2 Stratégie de compression

Nous conservons l'idée générale des techniques de codage de seconde génération, qui consiste à considérer l'image plutôt comme un ensemble de différentes structures que comme une matrice de pixels. Cette idée est adaptée aux caractéristiques des images ultrasons. Nous isolons donc tout d'abord les différentes structures pour ensuite coder l'information que chacune contient d'une façon adaptée. Cette idée fut présentée par Redfern et Hines dans [24] sous le nom de "a priori compression techniques". Leur stratégie consiste à ne considérer que l'information qui est utile. Afin de cerner cette information utile, la réduction de la résolution et de la dynamique

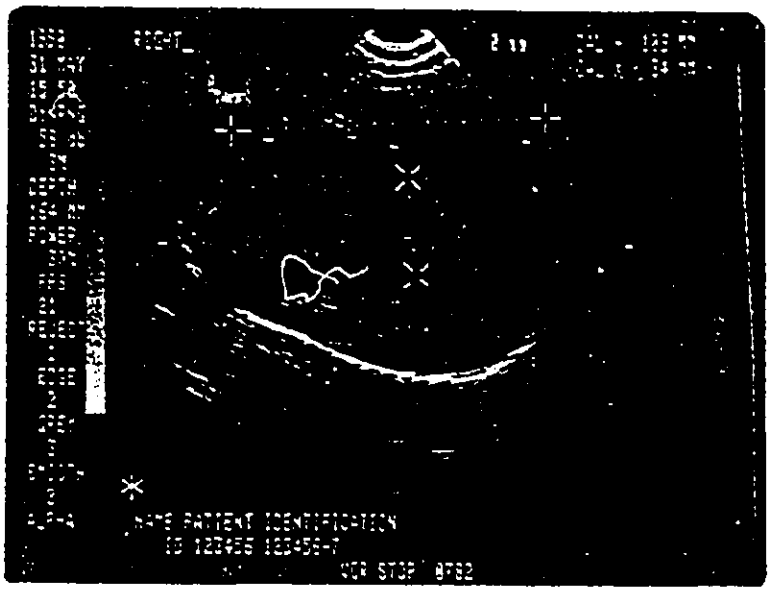

Image UM4 typique

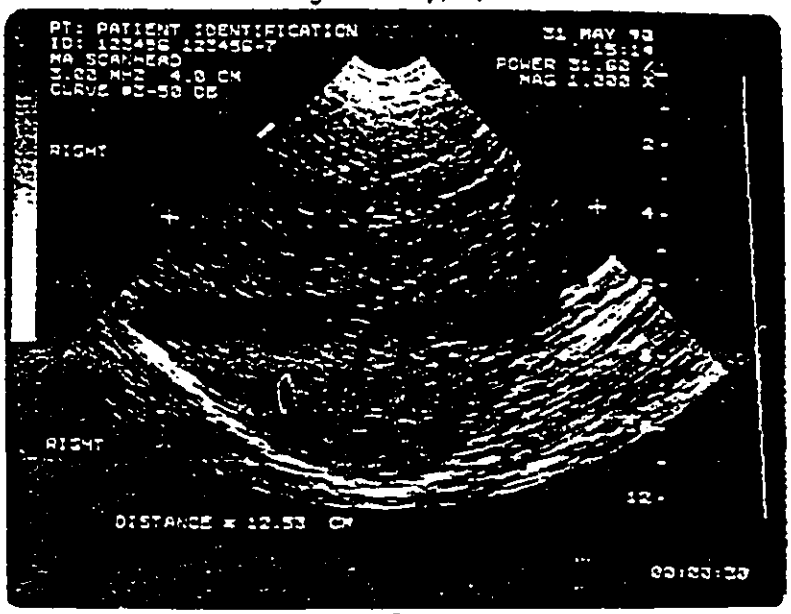

Image UM8 typique

Figure 4.4: Images ultrasons typiques

de l'image fut considérée, ainsi que la restriction de l'espace à coder aux régions d'intérêt. Nous nous limiterons à la restriction de l'espace, auquel nous ajouterons un traitement adapté de chacune des régions aux caractéristiques de l'information qu'elles contiennent. Nous devons donc premièrement segmenter notre image pour en extraire les différentes régions. Puis, nous coderons chacune d'entre elles selon le type d'information qu'elles contiennent.

Les figures 4.5 and 4.6 présentent une représentation schématique des deux types (UM4 et UM8) d'images ultrasons recensées. Ces représentations schématiques regroupent les

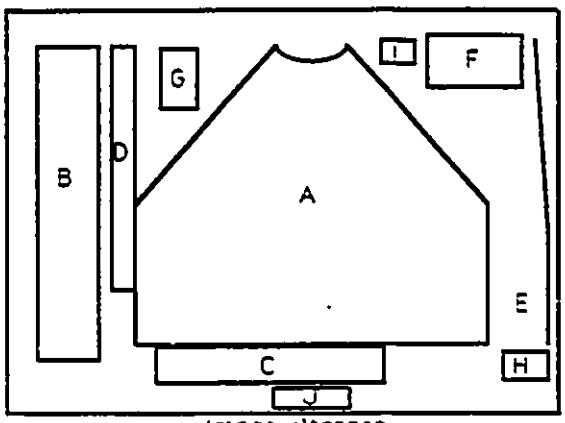

Image ultrason
type UM4

Bloc	Information
A	Image ultrason
B	Paramètres de l'image
C	Identification du patient
D	Echelle des niveaux de gris
E	Echelle de correction d'intensité
F	Position des pointeurs
G	Autres paramètres
H	" "
I	" "
J	" "

Figure 4.5: Représentation schématique des images ultrasons UM4

différentes informations sous une forme compacte, ce qui simplifie la discussion. Chaque bloc correspond à un groupe d'information de l'image. Nous pouvons classer les informations présentes dans l'image selon le modèle que nous utilisons pour les représenter. Nous pouvons classer ces informations selon trois modèles de représentation : le niveau de gris du pixel, le

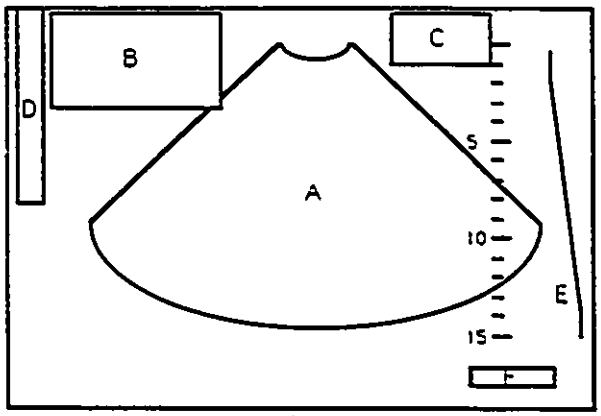

Image ultrason
type UM8

Bloc	Information
A	Image ultrason
B	Parametres de l'image/ID du patient
C	
D	Echelle des niveaux de gris
E	Echelle de corredtion d'intensité
F	Autres parametres

Figure 4.6: Représentation schématique des images ultrasons UM8

caractère imprimé, et l'expression mathématique. Nous classons chaque bloc de façon à ce que la modélisation de son information n'apporte aucune perte de son information. L'information de l'image ultrason et de l'échelle de gris est contenue dans le niveau de gris de chacun de ses pixels. Ces deux blocs sont donc classés sous le modèle du niveau de gris du pixel. L'information de tous les blocs de paramètres de création de l'image et d'identification du patient est contenu dans les caractères imprimés qu'ils contiennent. Ceux-ci sont donc classés sous le modèle du caractère imprimé. Finalement, l'information de l'échelle de correction d'intensité est contenue dans la forme de la droite, ce qui peut se représenter par une expression mathématique. Elle est donc classée sous le modèle de l'expression mathématique. Cette série de modèles ne nous permet pas cependant de représenter les pointeurs et demande un traitement poussé pour pouvoir tenir compte de la présence ou de l'absence de certains blocs. De plus la représentation de l'information sous le modèle des caractères imprimés demande un algorithme de reconnaissance de caractères.

Une autre série de modèles plus simples peut être utilisée pour classer nos informations soit : le niveau de gris des pixels, et la binarisation de l'intensité des pixels. Ici l'image ultrason et l'échelle de gris sont classées sous le modèle du niveau de gris du pixel et tous les autres blocs soit les caractères imprimés, l'échelle de correction d'intensité, et les pointeurs sont classés sous le modèle du niveau binaire du pixel. Cette classification de l'information est de niveau inférieur à la précédente car nous n'adressons pas l'information sous son plus haut niveau de représentation. Cependant elle ne nécessite pas un traitement aussi approfondi de l'information. Il est évident que la première série de modèles offrait une possibilité d'un plus haut taux de compression. En comparant simplement le codage d'un caractère imprimé qui ne demanderait que 7 bits sous la première série, celui-ci demanderait 128 bits sous la deuxième série de modèles. La simplicité de l'algorithme nous est apparu un aspect plus attrayant, et ce spécialement pour les pointeurs et l'échelle de correction d'intensité.

Le modèle de niveau de gris est le plus coûteux en fait d'espace mémoire. Il est donc nécessaire de restreindre le plus possible les régions où ce modèle sera utilisé. C'est ce que nous tenterons de faire en cherchant le plus précisément possible les limites de l'image ultrason

et de l'échelle de gris.

Finalement, étant donné que nous avons deux types d'images (UM4 et UM8) aux caractéristiques bien différentes, afin de simplifier leur traitement nous assumerons que le type d'image est connu a priori.

Chapitre 5

Segmentation de l'Image Ultrason

Comme nous l'avons établi au chapitre précédent, notre problème de compression de l'information des images ultrason se divise en deux parties ; la segmentation de l'image en trois régions et le codage de l'information contenue dans chacune de ces régions. Nous traiterons d'abord ici de la segmentation alors que le codage sera l'objet du prochain chapitre.

Quatre images ont été sélectionnées comme images tests. Le choix de celles-ci fut guidé par un souci de représentation des différentes formes de la région de "l'image ultrason" rencontrées et des deux types d'images soit UM4 et UMS. Ces quatre images sont présentées aux figures 5.1, 5.2, 5.3, et 5.4.

Les trois régions à extraire sont : "l'échelle de gris", "l'image ultrason" et "l'image de fond" qui elle comprend les caractères imprimés, les pointeurs, et l'échelle de correction d'intensité. L'approche générale consiste à extraire les régions une à une et à éliminer l'espace qu'elles occupent des zones de recherche des autres régions. L'ordre dans lequel nous abordons la segmentation des trois régions peut se révéler d'une certaine importance. Notons d'abord que la région de "l'image de fond" consiste en ce qui reste de l'image après l'extraction de la région de "l'échelle de gris" et la région "l'image ultrason". Il est donc logique d'extraire cette région par défaut après avoir extrait les deux autres. Les connaissances que nous avons des caractéristiques de l'échelle de gris soit sa position dans l'image, ses dimensions et son intensité relative, nous permettent d'extraire cette région relativement facilement. Ceci nous

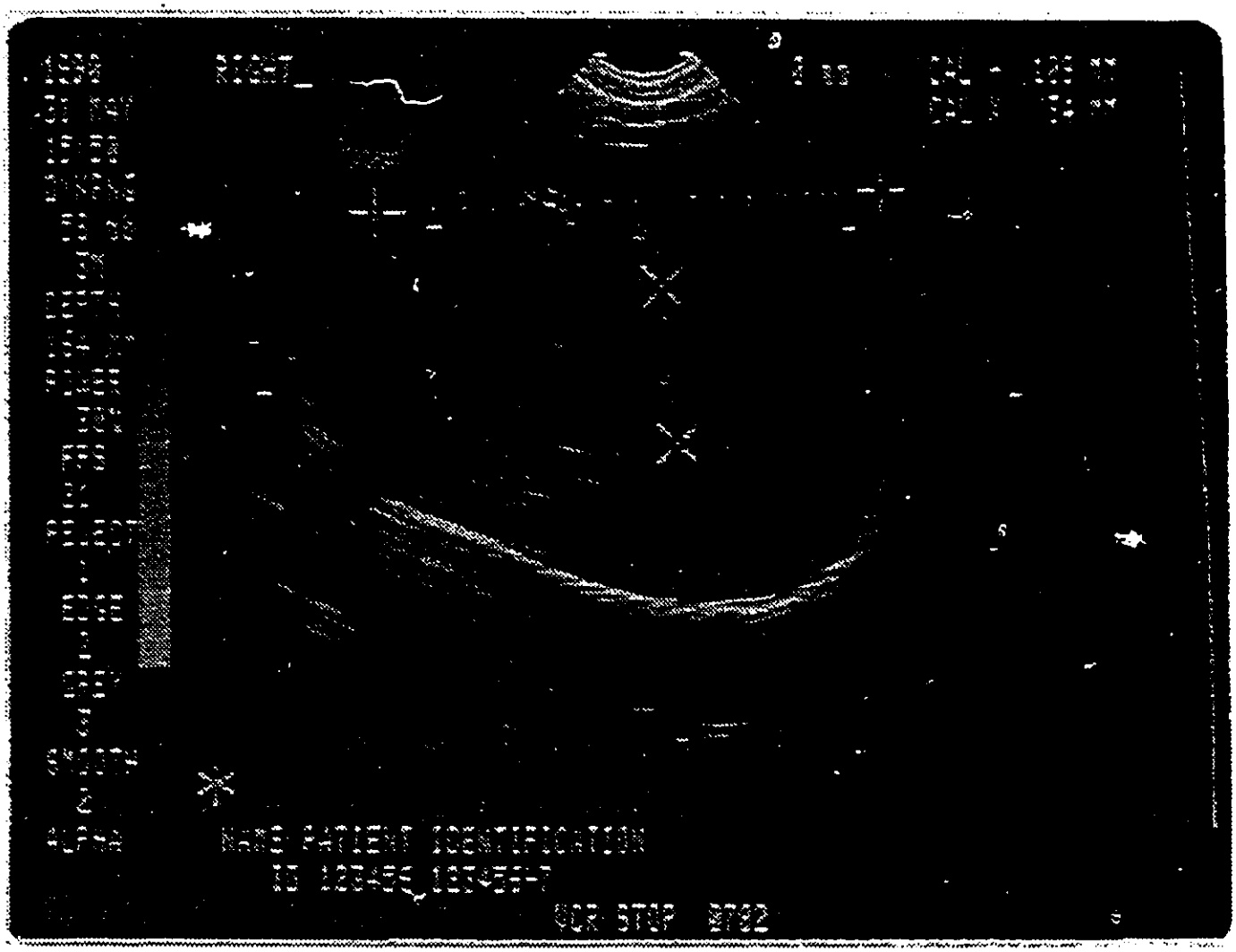

Figure 5.1: Image test "im4-11.ima"

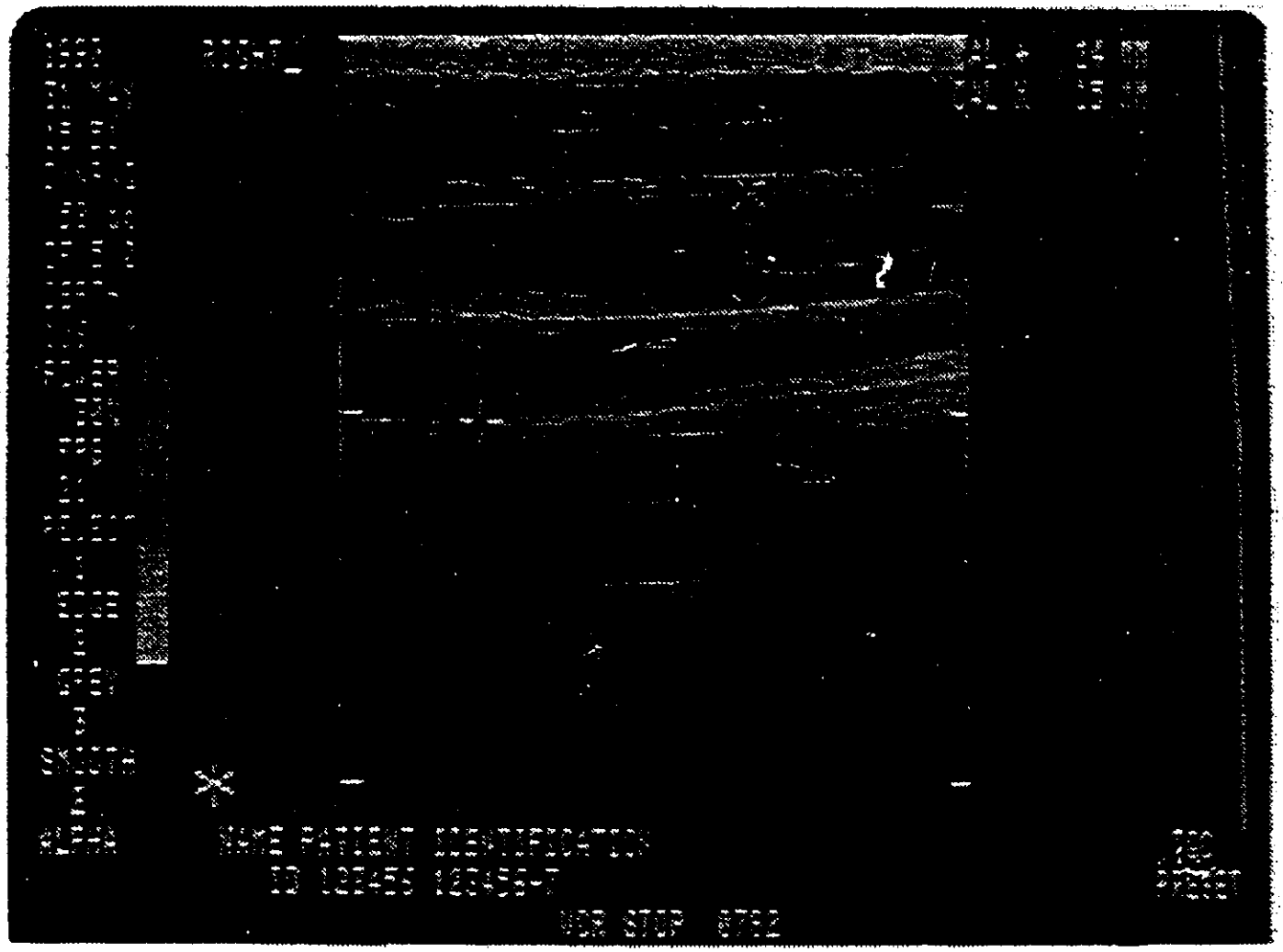

Figure 5.2: Image test "im4-14.ima"

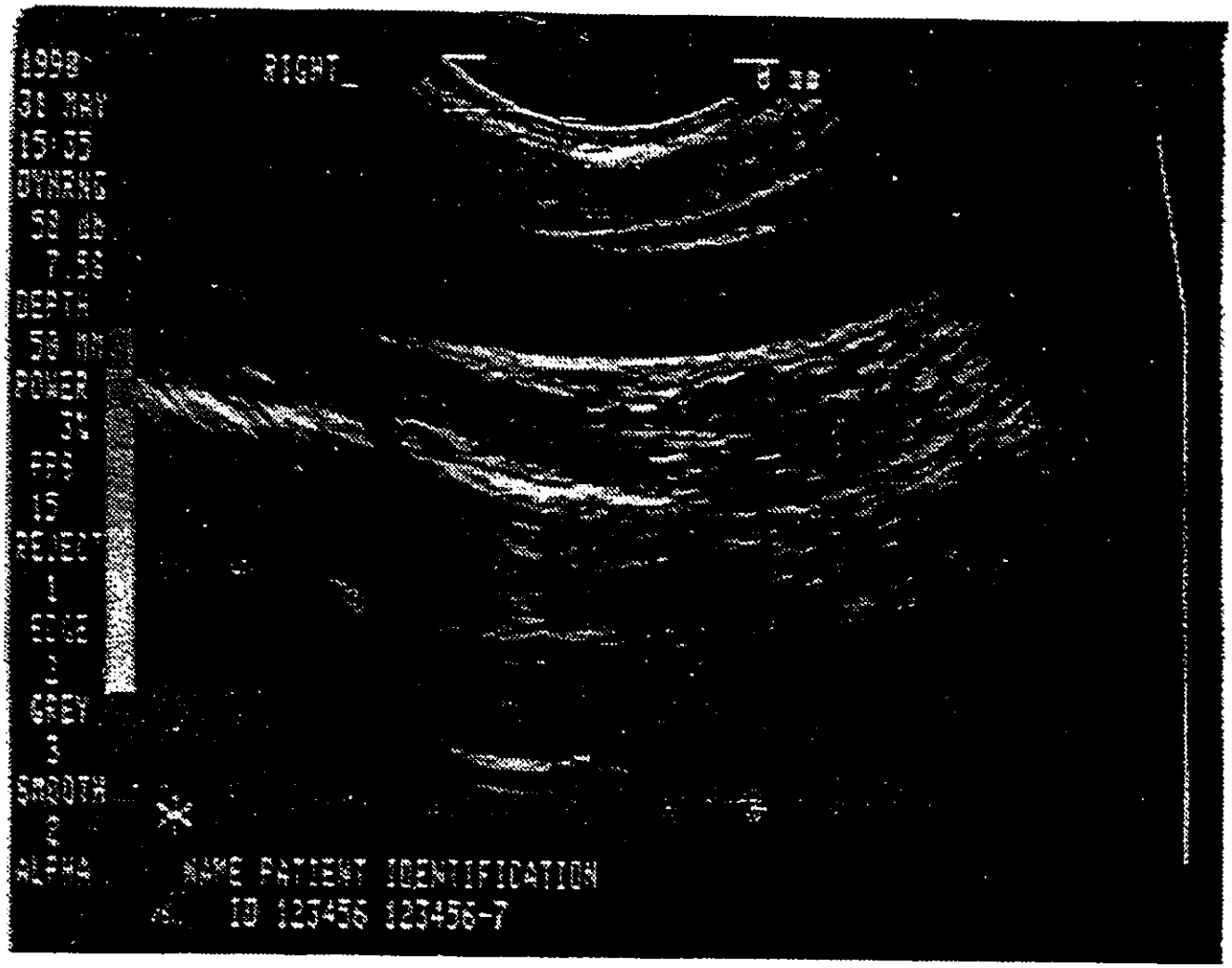

Figure 5.3: Image test "im4-25.ima"

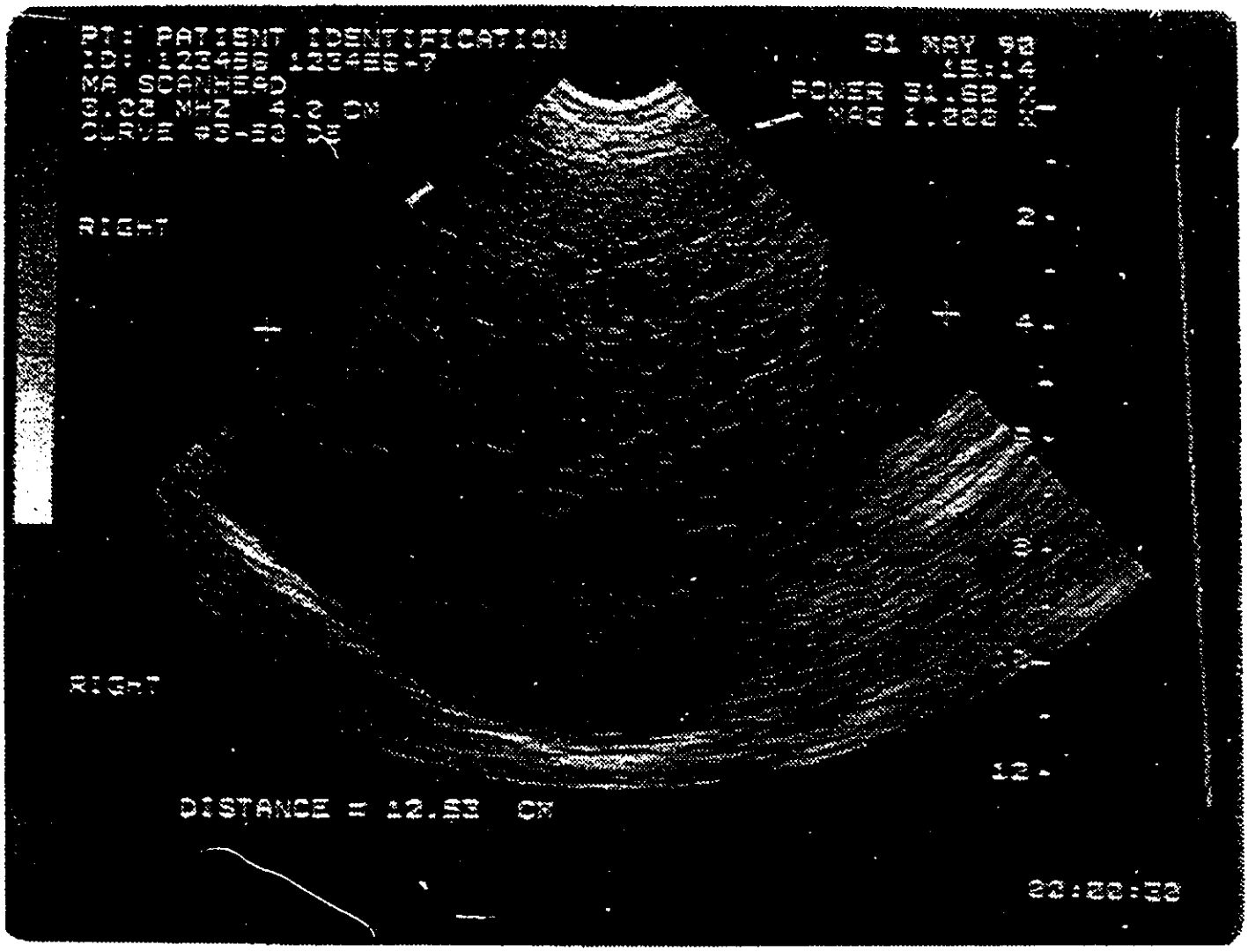

Figure 5.4: Image test "im8-11.ima"

permet alors de restreindre notre zone de recherche pour l'extraction beaucoup plus complexe de la région de "l'image ultrason". L'ordre adopté pour notre segmentation est donc "l'échelle de gris", "l'image ultrason" et "l'image de fond". Le schéma général de la segmentation de ces trois régions est présenté à la figure 5.5.

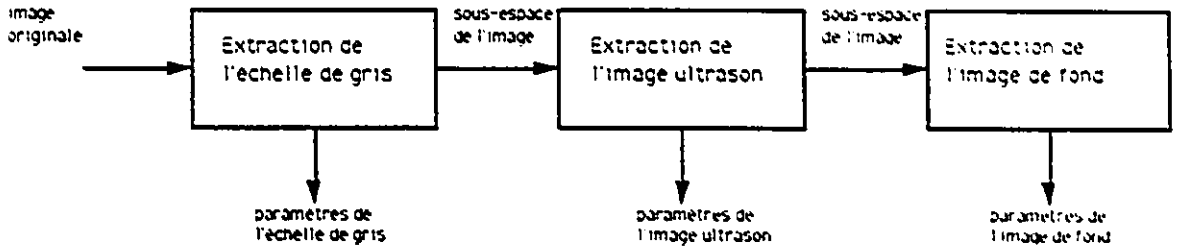

Figure 5.5: Etapes de la segmentation des images ultrason

Examinons maintenant les algorithmes de segmentation que nous avons développés pour chacune d'elles.

5.1 Segmentation de l'échelle de gris

Pour chaque type donné d'image, nous connaissons les dimensions de l'échelle de gris, et sa position approximative. Cette connaissance des dimensions, associée à la position précise d'un des coins de l'échelle de gris, nous permettrait de déduire ses autres limites. Le bas de l'échelle de gris par son intensité bien contrastée avec le fond de l'image présente un intérêt particulier pour la détection d'un tel point. Connaissant la position approximative de l'échelle de gris, il nous est possible d'établir une zone de recherche dans laquelle nous retrouverons assurément le bas de l'échelle de gris. Les limites de cette zone pour chaque type d'image sont données dans le tableau 5.1.

Le seuillage est une méthode bien adaptée à ces cas où les régions à extraire sont plutôt homogènes et bien contrastées. Le seuil peut être déterminé à l'aide de l'histogramme de l'intensité des pixels de cette zone recherche. Le seuil est choisi en évaluant l'importance des différentes vallées présentes dans l'histogramme. L'importance d'une vallée est évaluée en

Limites de la zone de recherche	*Type 1*	*Type 2*
limite gauche	35	0
limite droite	115	45
limite supérieure	305	240
limite inférieure	400	325

Table 5.1: Position des limites de la zone de recherche

calculant la différence entre la valeur du plus faible de ses deux maximums adjacents et la valeur du minimum de la vallée. Un exemple est présenté à la figure 5.6.

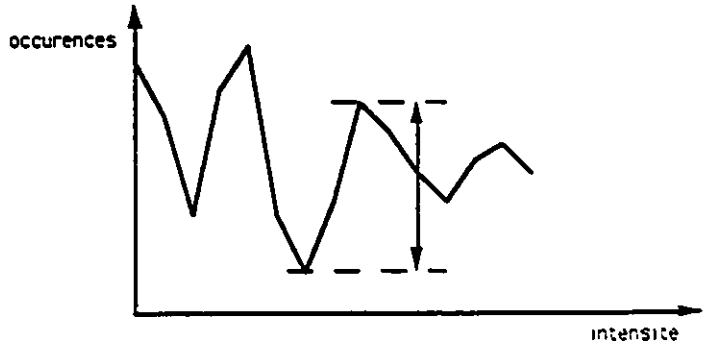

Figure 5.6: Mesure de l'importance d'une vallée

Ce critère ne prend cependant pas en considération la largeur des vallées. Pour remédier à cette situation, nous ne considérons que les vallées de largeur de plus de 2. Toute vallée ayant une largeur de seulement 2 verra la valeur de son minimum remplacée par la valeur de son plus faible maximum adjacent. Un exemple est présenté à la figure 5.7.

Malheureusement, le nombre de vallées est habituellement encore très élevé. Avant de passer à l'évaluation de l'importance des vallées nous désirons réduire ce nombre. Pour ce faire, nous exécutons une opération de lissage de l'histogramme. Cette opération consiste en une convolution de l'opérateur suivant sur l'histogramme.

$$histadouc[p] = (0.5\ hist[p-2] + hist[p-1] + 2\ hist[p] + hist[p+1] + 0.5\ hist[p+2])/5 \cdot$$

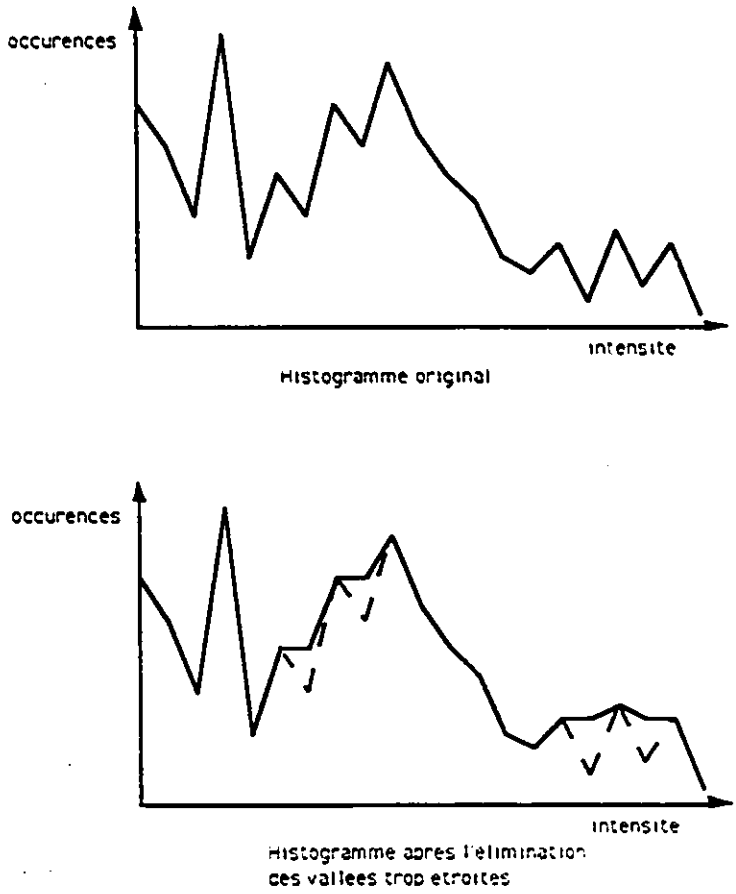

Figure 5.7: Elimination des vallées de largeur de moins de 2

L'histogramme est itérativement lissé jusqu'à ce qu'il ne contienne que 2 vallées. La position du minimum de la vallée la plus importante est choisie comme valeur du seuil. La zone de recherche est ensuite seuillée en fixant à "0" les pixels d'intensité inférieure à ce seuil et à "255" les pixels d'intensité supérieure. Nous obtenons ainsi une image similaire à celle présentée à la figure 5.8.

Figure 5.8: Image avec zone de recherche seuillée

La zone de recherche contient l'échelle de gris, et possiblement des caractères imprimés et une partie de l'image ultrason. Nous pouvons reconnaître l'échelle de gris par sa structure. Elle est constituée d'un bloc rectangulaire de pixels blancs. Il s'agit donc de reconnaître une suite consécutive de "C" colonnes de longueur minimale "L". Les valeurs "C" et "L" sont fixées selon le type d'image. Elles sont présentées au tableau 5.2.

La position de la première colonne de cette suite est choisie comme première approximation de la limite gauche. La limite verticale inférieure peut alors être déterminée en examinant à

Paramètres de segmentation	*Type 1*	*Type 2*
C	7	10
L	15	10
largeur maximale	20	25
hauteur standard	361	300

Table 5.2: Paramètres de segmentation selon le type d'image

quelle ligne cette suite de colonnes se termine. C'est sur cette ligne que l'échelle de gris atteint son intensité la plus forte, donc un plus fort contraste avec le fond de l'image. La confiance sur cette ligne, quant à l'appartenance de chacun de ces points, y est donc maximum. C'est pourquoi il est préférable d'y redéfinir les limites horizontales. La limite gauche est donc fixée à la position de la première colonne de la suite sur cette ligne. Due à la disposition des régions la limite droite est succeptible d'être confondue avec une structure interne de "l'image ultrason". Nous tentons quand même une recherche de celle-ci en identifiant la position où se termine la suite de colonnes. Cependant si celle-ci implique une largeur de l'échelle, supérieure à la largeur espérée, nous fixons la limite à la valeur de la limite gauche additionnée de la largeur standard pour ce type d'image. La limite verticale supérieure est fixée en soustrayant la valeur de la hauteur standard de la valeur de la limite verticale inférieure.

5.2 Segmentation de l'image ultrason

Nous ne connaissons pas la position exacte de la région de "l'image ultrason", mais nous savons qu'elle se trouve toujours à droite de l'échelle de gris. L'extraction de l'échelle de gris nous permet donc de déterminer une zone de recherche qui est limitée, à gauche, par la limite droite de l'échelle et qui est limitée dans les autres directions par les limites originales de l'image.

Cette zone de recherche comprend "l'image ultrason" et une partie de "l'image de fond". La région de "l'image ultrason" est constituée de deux types de groupes de pixels: des groupes

de pixels d'intensité moyenne élevée et comportant des variations importantes d'intensité , et des groupes de pixels d'intensité moyenne faible et ne comportant que très peu de variations d'intensité . De plus, sur l'ensemble des images observées, ce deuxième type de groupe de pixels comporte localement une intensité moyenne toujours plus faible que l'intensité moyenne de "l'image de fond".

"L'image de fond" semble globalement uniforme, ce qui nous laisserait croire à une identification facile de ses pixels par une recherche de l'intensité la caractérisant. Cependant cette région présente des variations importantes d'intensité sur l'ensemble de l'image et également une certaine activité locale. A l'échelle du pixel les deux régions comportent des similitudes qui ne nous permettent pas de les distinguer. De plus, pour les groupes de pixels comportant des variations importantes d'intensité , seule la considération d'une propriété locale permet de les relier en une même région. Il est donc nécessaire d'avoir une vision plus globale en faisant appel à une mesure de texture.

5.2.1 Recherche d'une mesure de texture

Une mesure de la texture en chaque point est obtenue en appliquant un opérateur qui extrait une valeur représentant une propriété sur l'ensemble des points d'un voisinage restreint. Le voisinage est habituellement défini comme étant une fenêtre carrée centrée sur le pixel d'intérêt. Une transformée de l'image est ainsi obtenue en remplaçant l'intensité de chaque pixel par la valeur de cette mesure. Rosenfeld et Kak dans [27], nous suggèrent de suivre cette opération par un lissage quelconque afin de réduire les variations ponctuelles.

Plusieurs opérateurs simples ont été évalués afin d'en choisir un qui nous permettrait de classifier facilement les pixels. Le critère de Fisher [6] fut utilisé pour évaluer les résultats des différents opérateurs. Il se définit comme suit :

$$f = (\eta_1 - \eta_2)^2/(\sigma_1^2 + \sigma_2^2) \tag{5.1}$$

où η_i est la moyenne des intensités des pixels de la région "i" et σ_i est la variance. Ce critère donne une mesure de la distance entre les distributions des deux régions. Il est donc nécessaire d'établir une zone représentative de chaque région. Nous avons choisi dans l'image "im4-11.ima" une zone de 87000 pixels pour la région de "l'image ultrason" et une zone de 11250 pixels pour la région de "l'image de fond".

Les différents opérateurs évalués sont les suivants :

1. Moyenne locale
2. Filtre médian
3. Intensité maximale - Intensité minimale
4. Valeur moyenne du Sobel
5. Variance
6. "Run-length"

Chacun de ces opérateurs, à l'exception de "Run-length", fut évalué pour des fenêtres de dimension variant de 3x3 à 9x9 pixels. L'opérateur "Moyenne locale" calcule la moyenne locale de l'intensité des pixels. L'opérateur de "Filtre médian" est un simple filtre médian. L'opérateur "Intensité maximale - Intensité minimale" calcule la différence entre la valeur du pixel d'intensité maximale et la valeur du pixel d'intensité minimale du voisinage. L'opérateur "Valeur moyenne du Sobel" calcule la valeur moyenne locale du Sobel de l'image. L'opérateur "Variance" calcule la variance de l'intensité des pixels du voisinage. L'opérateur "Runlength" est une variante d'un opérateur présenté par Haralick dans [9] dont nous avons parlé à la section 2.4.2. L'opérateur évalue pour chaque point le nombre de pixels qui, selon une direction donnée, restent à l'intérieur d'un seuil de variation d'intensité fixé. La valeur la plus importante du nombre de pixels à l'intérieur de ce seuil est choisie comme nouvelle valeur du pixel. Cet opérateur fut évalué pour des seuils variant de 4 à 11 niveaux de gris et

avec deux directions (verticale et horizontale) et quatre directions (verticale, horizontale, et les deux diagonales). Les résultats de cette évaluation sont présentés aux figures 5.9, 5.10.

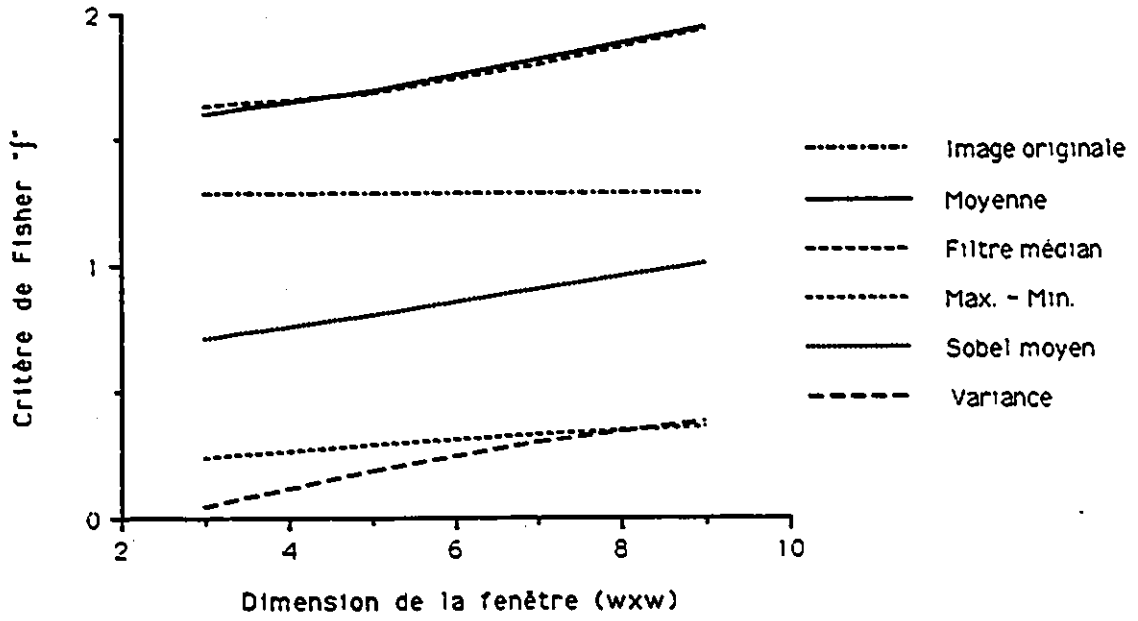

Figure 5.9: Evaluation des mesures de textures

Les résultats démontrent clairement la supériorité de l'opérateur "Runlength". Nous avons choisi celui de deux directions avec un seuil de "8" niveaux de gris. Ce choix fut le fruit d'un compromis entre la séparabilité des régions et l'érosion de la région d'intérêt. En effet, un seuil plus élevé avait tendance à éroder la région d'intérêt, bien qu'il polarisait davantage l'intensité des pixels des deux régions. Le choix de l'opérateur "Runlength" selon deux directions plutôt que celui selon quatre directions fut motivé par la même raison. L'opérateur de lissage que nous avons choisi est un filtre médian 3x3. Le filtre médian a l'avantage de préserver les limites abruptes. A la figure 5.10 nous pouvons constater l'effet du lissage sur le critère de Fisher.

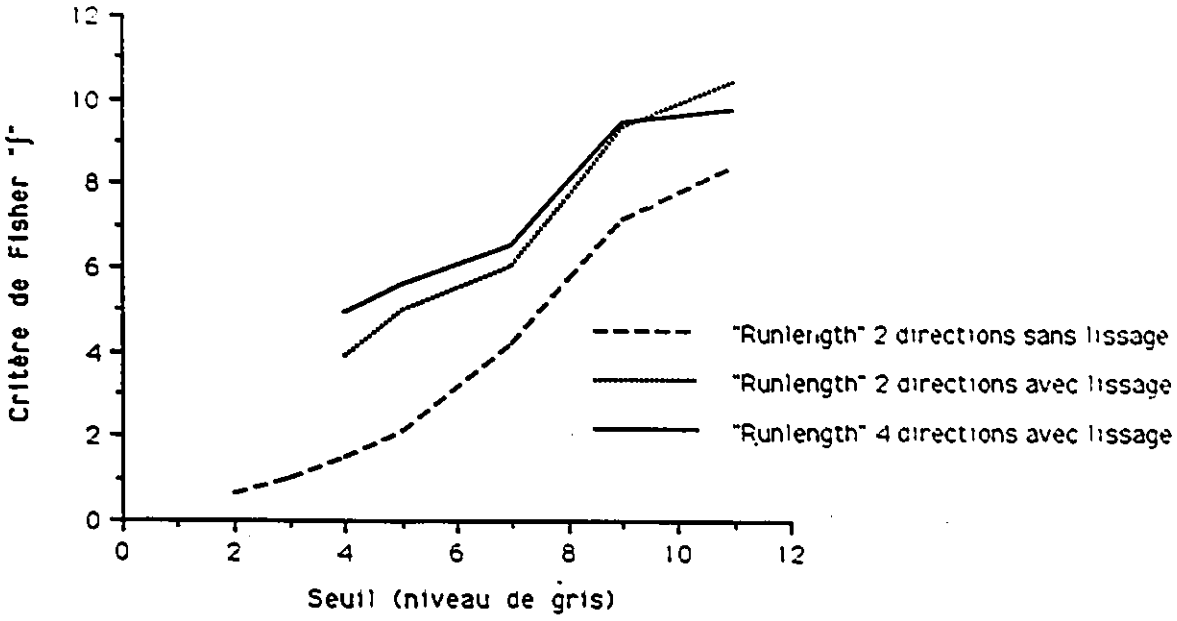

Figure 5.10: Evaluation des mesures de textures

5.2.2 Première segmentation

Ces opérations nous procurent une nouvelle image constituée de zones de faible intensité là où l'activité du signal est importante et de zones de forte intensité là où l'activité du signal est moins importante. Nous désirons extraire les zones de faible intensité. Elles contrastent assez bien avec le reste de l'image ce qui nous laisse croire qu'un simple seuillage pourrait suffire. L'observation de l'histogramme de l'image confirme cette hypothèse. Celui-ci comporte un "cluster" majeur à chaque extrémité de l'histogramme et quelques autres de moyenne importance entre ceux-ci. Cependant, la non-proximité des deux "clusters" ainsi que la présence des autres "clusters" nous compliquent la tâche quand au choix d'un seuil approprié. Après essais nous avons constaté que l'intensité correspondant à la fin du premier "cluster" constituait une valeur appropriée pour le seuil. Il est donc nécessaire de développer une méthode pour déterminer ce point. Nous notons que la dérivée seconde de l'histogramme lissé comporte toujours un maximum local en cette région. Deux maximums locaux sont

présents soit un à gauche du "cluster" et un à droite. Le maximum de droite est le seul qui nous intéresse. Il est donc tout d'abord nécessaire d'identifier le centre du "cluster" afin d'orienter nos recherches à sa droite. La position du centre peut être approximée par la position du maximum du "cluster".

Le lissage fut incorporé dans l'opérateur approximant la dérivée première et seconde. La dérivée première est approximée par l'opérateur suivant:

$$hist'[p] = (hist[p+3] + hist[p+2] + hist[p+1] - hist[p-1] - hist[p-2] - hist[p-3])/3$$

La dérivée seconde est approximée par l'opérateur suivant:

$$hist''[p] = (hist'[p+3] + hist'[p+2] + hist'[p+1] - hist'[p-1] - hist'[p-2] - hist[p-3])/3$$

Aux figures 5.11, 5.12 et 5.13 nous présentons respectivement un exemple d'un histogramme, de sa dérivée première et de sa dérivée seconde.

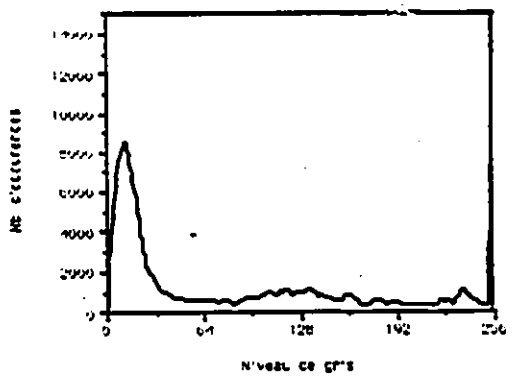

Figure 5.11: Histogramme lissé

Il ne nous reste qu' à identifier ce maximum de la dérivée seconde. Afin d'inclure la portion restante du "cluster" nous augmentons la valeur du seuil en fixant celui-ci à l'intensité de

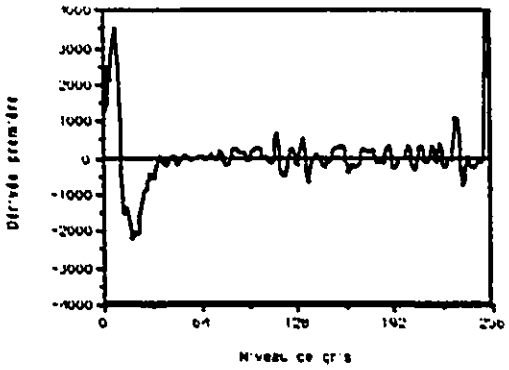

Figure 5.12: Dérivée première de l'histogramme

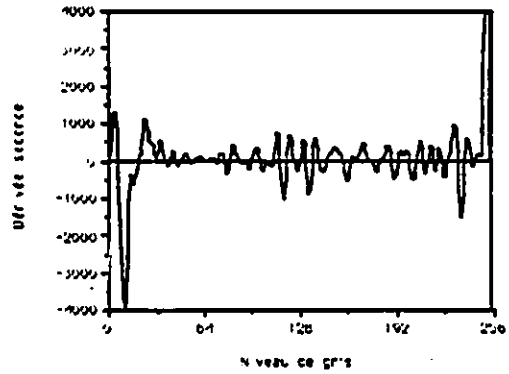

Figure 5.13: Dérivée seconde de l'histogramme

l'histogramme qui a une valeur de dérivée seconde correspondant à 10% de ce maximum. Nous seuillons donc l'image en fixant à "0" (noir) les pixels d'intensité supérieure au seuil et à "255" (blanc) les pixels d'intensité inférieure.

Extraction de la plus grande région

L'image que nous obtenons après seuillage comporte plusieurs régions qui ne font pas toutes partie de la région de "l'image ultrason". Cependant la plus grande de toutes ces régions en fait toujours partie. Nous allons donc l'identifier pour, par la suite, tenter d'en extraire les limites de la région de "l'image ultrason". Pour ce faire nous allons utiliser un algorithme de suivi de contour, puis un algorithme de remplissage de contour, afin d'établir le nombre de pixels appartenant à chacune de ces régions. Les deux algorithmes que nous avons choisis ont été utilisés par Cheng dans [3]. L'image est d'abord balayée ligne par ligne jusqu'à détection d'un pixel appartenant à une de ces régions. Les algorithmes de suivi de contour et de remplissage sont alors exécutés sur cette région.

Le suivi de contour débute par ce premier pixel. Celui-ci est identifié comme étant le pixel de départ du contour. La position du prochain pixel à être considéré dépend du statut du dernier pixel considéré et de la direction utilisée pour l'atteindre. Huit directions sont possibles (voir figure 5.14). Nous suivons les contours dans le sens horaire. Le tableau 5.2.2 présente la direction à suivre pour atteindre le prochain pixel à considérer en fonction du statut du dernier pixel considéré et de la dernière direction utilisée. Si le pixel considéré est un membre du contour (pixel blanc) nous nous repositionnons sur celui-ci, sinon nous restons à la position du dernier pixel du contour identifié. Il tient compte également du cas où cette direction nous amènerait à une position à l'extérieur de l'image.

L'algorithme de remplissage recense tous les pixels appartenant à la région du contour précédemment détecté. Pour ce faire il regroupe itérativement tous les pixels blancs adjacents aux pixels du contour, puis tous les pixels blancs adjacents à ces pixels adjacents et ainsi de suite jusqu'à ce qu'aucun pixel blanc ne soit adjacent à cette région en croissance. Le

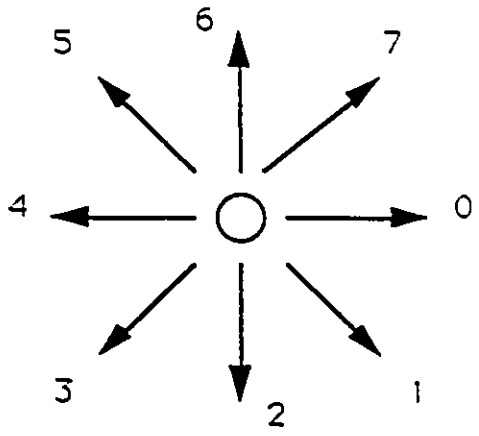

Figure 5.14: Directions possibles pour le suivi de contour

Dernière direction utilisée	*Dernier pixel testé n'appartenait pas à la région ou position à l'extérieur de l'image*	*Dernier pixel testé appartenait à la région*
0	1	5
1	2	6
2	3	7
3	4	0
4	5	1
5	6	2
6	7	3
7	0	4

Table 5.3: Direction pour le prochain pixel à testé

nombre de pixels de chaque nouvelle région ainsi extraite est comptabilisé. L'intensité de ces pixels est ensuite fixée à la valeur de l'intensité du fond afin qu'ils ne soient pas considérés de nouveau lors de l'extraction de la région suivante. La région comptant le plus grand nombre de point est conservée. Les régions ainsi isolées pour chaque image test sont présentées aux figures 5.15 à 5.18.

Figure 5.15: Plus grande région pour l'image "im4-11.ima"

5.2.3 Extraction des limites

En isolant ces régions nous avons extrait la majeure partie des groupes de pixels du premier type appartenant à la région de "l'image ultrason". Nous allons donc tenter d'en extraire les limites de la région de l'image ultrason. Pour ce faire nous allons exploiter la connaissance que nous avons des formes possibles de cette région. En effet, en limitant le nombre de formes ou plutôt en connaissant les formes susceptibles d'être rencontrées, il est possible de simplifier l'algorithme considérablement. Examinons d'abord les différentes formes que nous avons recensées. Quatre formes typiques ont été recensées et sont présentées à la figure 5.19.

Trois types de courbe constituent les limites de ces formes soit la droite verticale ou horizontale, la droite inclinée, et l'arc de cercle. Nous catégorisons ces formes en deux classes:

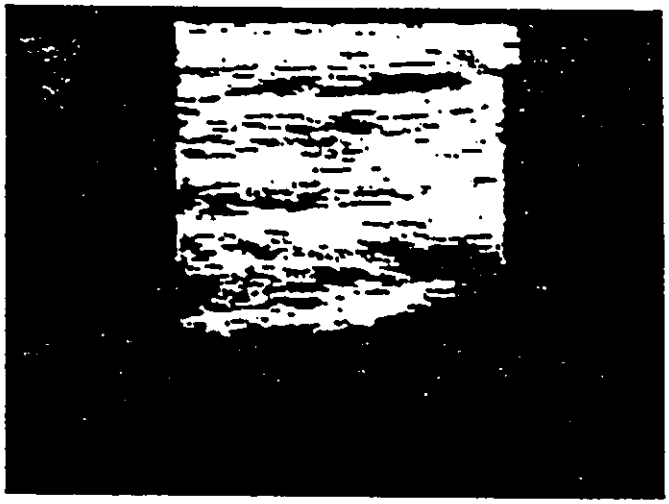

Figure 5.16: Plus grande région pour l'image "im4-14.ima"

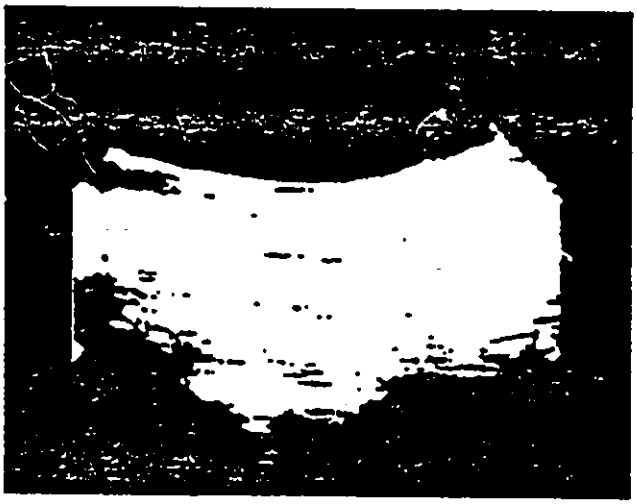

Figure 5.17: Plus grande région pour l'image "im4-25.ima"

Figure 5.18: Plus grande région pour l'image "im8-11.ima"

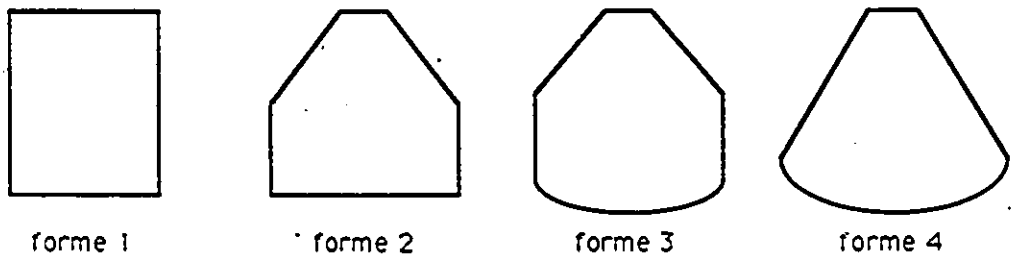

Figure 5.19: Formes recensées pour la région de "l'image ultrason"

formes sans limites inclinées (forme 1), et formes avec limites inclinées (formes 2, 3 et 4). Nous les catégorisons de cette façon, car une fois les limites constituées de droite inclinée extraites, l'algorithme devient le même pour toutes les formes, soit la détection d'une limite supérieure et inférieure verticalement, et d'une limite gauche et droite horizontalement. Nous définissons tout d'abord des limites approximatives de la région soit la position des extrémités gauche, droite, supérieure et inférieure. Nous classons alors la région extraite, en comparant la largeur de la région extraite dans sa partie supérieure avec sa largeur maximale. Si la largeur moyenne des 15 premières ligne de la région est de beaucoup inférieure à la largeur maximale, soit inférieure à 11/16 de la valeur maximale, nous considérons que nous avons affaire à une forme avec limites inclinées.

Limites de droites inclinées

Si nous avons des limites inclinées, nous allons d'abord les extraire. Pour ce faire, nous utilisons la transformée de Hough. Les points utilisés pour celle-ci sont extraits en balayant l'image ligne par ligne et en conservant les premiers points de chaque ligne pour la limite de gauche et les derniers de chaque ligne pour la limite de droite. Il est nécessaire de ne balayer que les lignes correspondant à la zone des limites inclinées. Nous ne balayerons donc que de la première ligne de la région à la ligne où la largeur de la région devient proche de sa largeur maximale, soit supérieure à 9/10 de la largeur maximale.

La représentation normale d'une droite est utilisée pour le calcul de la transformée de Hough.

$$\rho = x\cos\theta + y\sin\theta \tag{5.2}$$

Pour tous les points, la valeur de ρ est calculée pour des angles susceptibles d'être ceux de la droite recherchée, soit de 0.35 à 1.2 radian avec un incrément de 0.0025 radian. La valeur calculée de rho est approximée au 0.25 pixel près. Les valeurs admises de rho vont de 175 à 475. L'espace des paramètres θ et ρ est divisé en cellules de dimension correspondant

à la valeur de l'incrément de l'angle et de la précision sur ρ. Nous avons donc besoin d'une matrice de 340 par 1200 cellules, chaque cellule représentant une combinaison d'un angle et d'une valeur de rho. Les cellules sont tout d'abord initialisées à zéro. Pour chaque valeur de ρ calculé , la valeur de la cellule correspondant à la combinaison de ρ et de θ est incrémentée. Notre matrice permettant une grande précision pour le calcul de la valeur de rho, les points sensiblement sur une même droite n'incrémenteront pas une seule cellule mais plutôt un groupe de cellules. Cet étalement des positions des incrémentations dans la matrice peut être compensé en examinant les cellules par groupe de cinq de la façon présentée à la figure 5.20. Le groupe de cellules dont la somme des valeurs est la plus élevée est retenu.

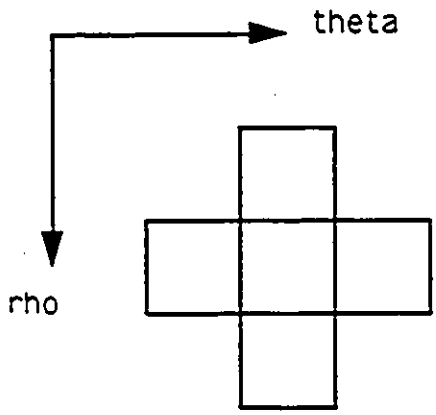

Figure 5.20: Groupe de cinq cellules examinées

Si plusieurs groupes de cellules correspondent à cette valeur maximale, nous augmentons la résolution en n'examinant que trois des cellules de chaque groupe comme indiqué à la figure 5.21. Afin de profiter de toute la précision sur la valeur de ρ et de θ que la matrice nous permet, nous choisissons les paramètres ρ et θ de la cellule centrale du groupe.

Avant d'accepter ces paramètres comme étant représentatifs de la limite recherchée, il est nécessaire d'évaluer leur fiabilité. Celle-ci peut être évaluée en calculant le nombre de points associés à cette droite. Nous avons fixé une limite minimale de 20 points qui doit être atteinte à résolution réduite sur trois cellules (voir figure 5.21). Si cette limite n'est pas atteinte, nous faisons appel à une étape de deuxième segmentation afin de compléter la région précédemment extraite. Nous traiterons de cette deuxième segmentation dans la

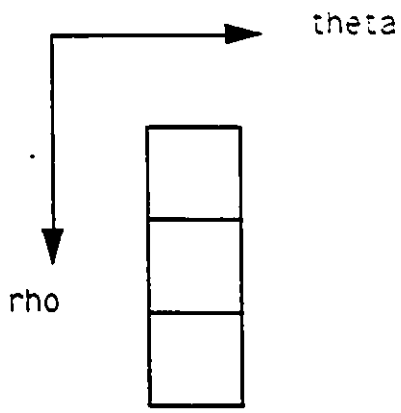

Figure 5.21: Groupe de trois cellules examinées

prochaine section. La limite supérieure pour les formes avec droites inclinées est déterminée par le point de rencontre des deux droites inclinées si ce point est à l'intérieur de l'image, sinon la limite est fixée à la première ligne de l'image.

Limites verticales

Les limites verticales, gauche et droite peuvent être extraites plus facilement en assumant que les points limites de celles-ci se situent tous sur une même ligne verticale. L'image est donc balayée ligne par ligne afin de trouver la position du premier et du dernier pixel de chaque ligne appartenant à la région. Afin d'accélérer le processus et de ne pas extraire des points non-pertinents, nous limitons la zone de recherche. Nous utilisons les limites approximatives pour fixer celles-ci à l'exception de la limite supérieure. Cette dernière est fixée à la position verticale du point d'intersection de la droite inclinée avec la limite horizontale de la zone de recherche. Les positions des premiers et des derniers pixels sont donc comptabilisées dans deux vecteurs. Ceux-ci sont d'abord initialisés à zéro, et le contenu de la position du vecteur correspondant à la position de chaque premier pixel est incrémenté ainsi que le contenu de la position du vecteur correspondant à la position de chaque dernier pixel. La frontière sera habituellement bruitée, et de plus une légère inclinaison de l'image est possible, ce qui rend le recensement colonne par colonne peu précis. Une vision plus globale soit sur deux colonnes

à la fois nous permettrait de tenir compte de cette variance autour de la position de la limite recherchée. Pour chaque position extraite nous incrémentons donc le contenu correspondant à la position du pixel et le contenu de la position précédente pour le vecteur des premiers pixels et le contenu de la position suivante pour le vecteur des derniers pixels de chaque ligne. Le choix d'incrémenter le pixel précédant pour le cas des premiers pixels et du pixel suivant pour le cas des derniers pixels nous assure que si un déplacement de la limite est occasionné par ce processus il occasionnera un ajout à la région recherchée et non un manque. Une fois toutes les lignes d'intérêt examinées la position des vecteurs dont le contenu est le plus élevé est retenue.

Avant d'accepter ces positions comme étant les limites gauche et droite de notre région, nous évaluons leur fiabilité. Celle-ci est évaluée en comparant les valeurs des vecteurs retenues à un nombre minimal nécessaire. Ce nombre est fixé au tiers du nombre de lignes évaluées. Si ce nombre n'est pas atteint une deuxième segmentation est nécessaire.

Limite inférieure

Deux types de courbes peuvent constituer la limite inférieure, soit la ligne droite ou l'arc de cercle. Toutes deux seront considérées.

Pour le cas de la ligne droite, la stratégie pour identifier cette limite est très similaire à celle utilisée pour identifier les limites verticales. Les mêmes limites pour la zone de recherche sont utilisées. Un balayage vertical de l'image est effectué de bas en haut, afin d'identifier la position de la limite inférieure sur chaque colonne. Un vecteur sert à comptabiliser le nombre d'occurences en chaque position. Le même procédé est utilisé pour contrer l'effet de l'inclinaison et des frontières bruitées, soit l'incrémentation à la position du pixel et à la position suivante verticalement. La fiabilité est évaluée de la même manière en fixant le nombre nécessaire à un tiers des colonnes examinées.

Pour le cas de l'arc de cercle, nous utilisons la transformée de Hough adaptée à l'équation

du cercle. Les points utilisés pour celle-ci sont extraits en balayant l'image verticalement de bas en haut de la limite approximative gauche à la limite approximative droite et en conservant les coordonnées des premiers points appartenant à la région pour chaque colonne. Nous utilisons l'équation suivante pour la représentation de l'arc de cercle.

$$r = \sqrt{(y - y_c)^2 + (x - x_c)^2} \tag{5.3}$$

Pour tous les points, la valeur de r est calculée pour des coordonnées du centre du cercle $(x_c; y_c)$ susceptibles d'être celles du cercle recherché. On calcule d'abord les coordonnées du point de rencontre des deux droites inclinées, ce qui constitue une bonne approximation des valeurs des coordonnées du centre du cercle. Le rayon r est alors calculé pour une série de valeurs x_c et y_c de plus et moins 25 pixels, centrée sur ce point, avec un incrément de 0.5 pixel. La valeur calculée de r est approximée au 0.5 pixel près. On calcule également une valeur minimale espérée pour le rayon de la façon suivante :

$$r_{min} = limite\ \ inf\acute{e}r.\ \ approx. - m_c - 15 \tag{5.4}$$

où m_c est la coordonnée en x du point de rencontre des deux droites inclinées. Les valeurs admises de r sont fixées de r_{min} à r_{min}+300. L'utilisation de ces trois paramètres demanderait une matrice de trois dimensions de 600x100x100. Afin de réduire l'espace mémoire requis nous nous limiterons à calculer la valeur du rayon qu'en ne faisant varier qu'un des deux paramètres à la fois soit y_c, en fixant x_c. Une matrice 2-D de 600x100 sera ainsi produite de laquelle nous extrairons le groupe de cellules dont la somme des valeurs est la plus élevée (voir figure 5.22). De même si plusieurs groupes de cellules correspondent à cette valeur maximale, nous augmenterons la résolution en n'examinant que trois cellules de chaque groupe comme indiqué à la figure 5.23.

Nous faisons alors varier la valeur de x_c en l'incrémentant, et nous recommençons les mêmes étapes soit la création d'une matrice 2-D et l'extraction du groupe de cellules de la matrice dont la somme des valeurs est la plus élevée. Une fois toutes les valeurs de x_c examinées, nous conservons comme valeurs des paramètres celles correspondant à la valeur du groupe de cellules à basse résolution dont la somme des valeurs est la plus importante.

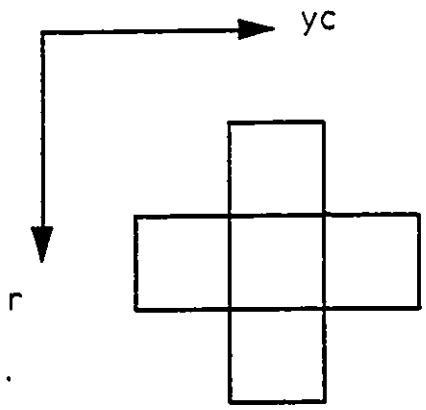

Figure 5.22: Groupe de cinq cellules examinées

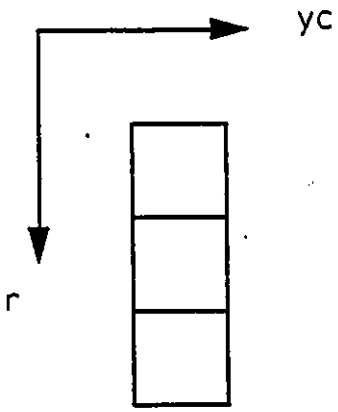

Figure 5.23: Groupe de trois cellules examinées

Avant d'accepter ces paramètres comme étant représentatifs de la limite recherchée, nous évaluons leur fiabilité. Les résultats sont dits fiables si plus de 1/7 des points évalués sont associés aux paramètres de cette équation.

Limites pour la forme 1

Si nous n'avons pas de limites inclinées nous avons nécessairement affaire à une région de forme 1. Pour ce type de forme nous utilisons les mêmes techniques pour la détection des limites verticales que celles décrites précédemment, et pour les limites horizontales la même technique que pour le cas de la ligne droite comme limite inférieure. Cependant le niveau de fiabilité exigé est différent. Pour les limites verticales et la limite supérieure aucun niveau de fiabilité n'est exigé. Pour la limite inférieure, le même niveau de fiabilité est exigé soit 1/3 des points évalués.

5.2.4 Deuxième segmentation

La première segmentation nous a permis de déterminer à quelle classe l'image appartient. De plus, si la qualité du résultat de cette segmentation le permet, des limites sont déterminées. Le résultat de cette première segmentation sert alors de structure de base sur laquelle on construira en ajoutant d'autres régions. La deuxième segmentation est effectuée afin d'identifier par ordre les régions susceptibles d'être ajoutées à la structure de base. Un processus itératif est alors mis en marche dans lequel après l'ajout de chaque nouvelle région, une recherche des limites non encore trouvées est effectuée et ce jusqu'à ce que toutes les limites soient détectées. Cet algorithme est présenté à la figure 5.24. Examinons maintenant chacune de ces étapes qui suivent la première segmentation d'une façon plus approfondie.

Tout d'abord l'image de départ est masquée à l'aide de l'image binaire obtenue après la première segmentation. Les pixels correspondant à la région extraite sont fixés à "255" et les autres pixels sont maintenus à leur valeur initiale. Ensuite un filtre médian de dimension

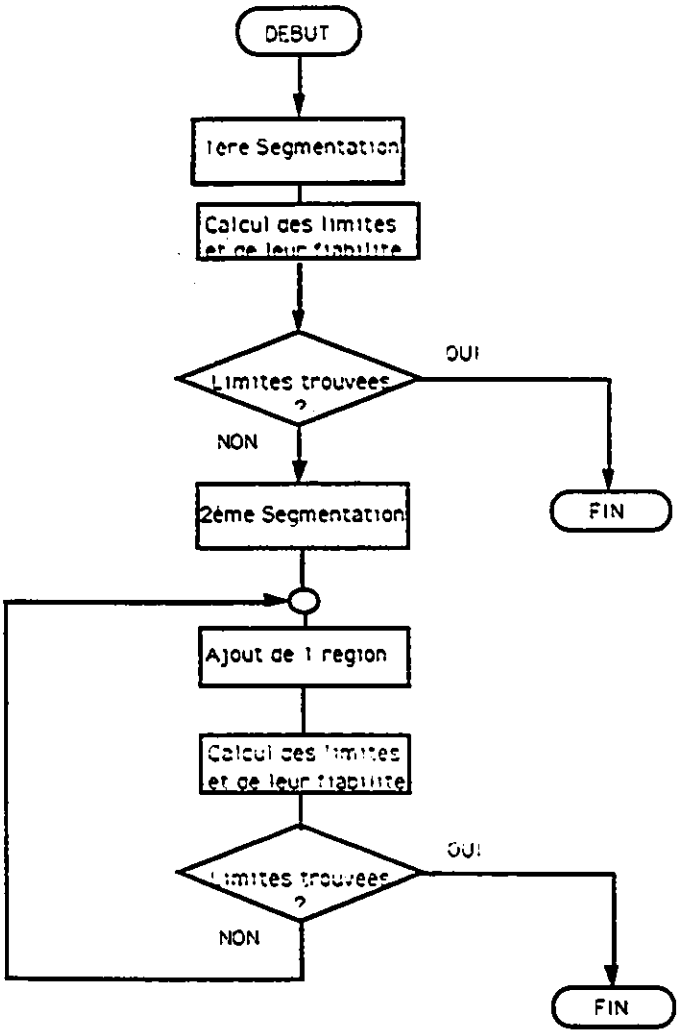

Figure 5.24: Algorithme de segmentation de la "région de l'image ultrason"

5x5 est appliqué sur l'image résultante afin d'éliminer les variations d'intensité ponctuelles dans l'image. Finalement, un algorithme de segmentation par fusion développé par Suk et Chung [31] est appliqué sur l'image. Celui-ci fut décrit à la section 2.1.3. Il consiste à examiner les pixels quatre par quatre et à évaluer la fusion formant le plus gros groupe de pixel rencontrant un critère d'homogénéité, soit la différence maximale d'intensité des pixels regroupés. Nous fixons cette différence à "2". La fusion de ces pixels peut impliquer la fusion de deux régions différentes. Il est donc nécessaire de fixer le critère contrôlant celles-ci. Il s'agit de la différence maximale entre les intensités moyennes de chaque région à regrouper. Nous fixons cette différence à "1".

Cet algorithme extrait un certain nombre de régions. Pour chaque région nous connaissons le nombre de pixels la composant et l'intensité moyenne de ceux-ci. Un processus itératif est alors mis en marche dans lequel nous fusionnons les régions une à la fois à la région extraite lors de la première segmentation, et ce jusqu'à ce que la région ainsi formée nous permet d'extraire toutes les limites. L'ordre dans lequel les régions sont ajoutées est fonction de leur intensité moyenne. Les régions sont ajoutées par ordre croissant de leur intensité moyenne car comme nous l'avons mentionné précédement, une fois le premier type de groupe de pixels extrait, le deuxième type de groupe de pixels se distingue de l'image de fond par leur intensité localement plus faible. Afin de ne pas considérer des régions non-pertinentes, nous ne considérons que des régions se situant à l'intérieur des limites approximatives augmentées d'une marge de 15 lignes ou colonnes ou à l'intérieur des limites déjà déterminées. L'algorithme de segmentation utilisé pour extraire ces régions produit plusieurs petites régions. Ceci occasionnerait un nombre excessif d'itérations qui aurait comme conséquence de ralentir considérablement le processus d'extraction des limites. Pour éviter ceci nous fixons une limite minimale sur la taille des régions considérées. Celle-ci est fixée à 15 pixels. Les figures 5.25 à 5.29 nous présentent l'évolution de la region après l'ajout de 0, 10, 20, 30, 39 régions.

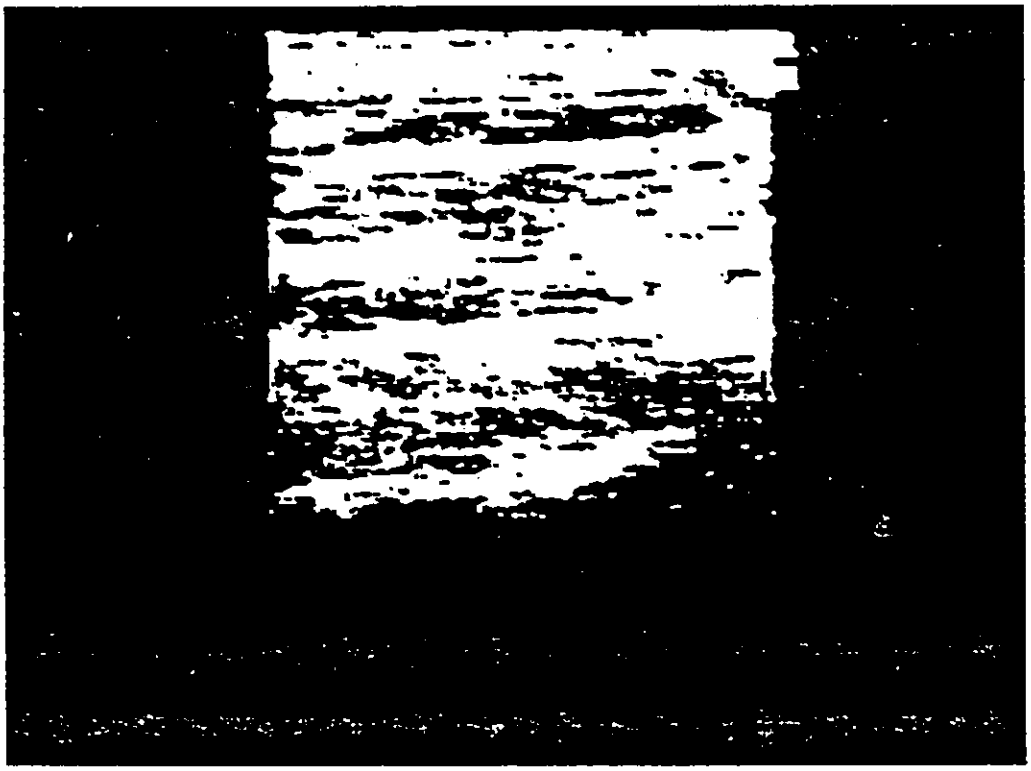

Figure 5.25: Région extraite après l'ajout de 0 régions

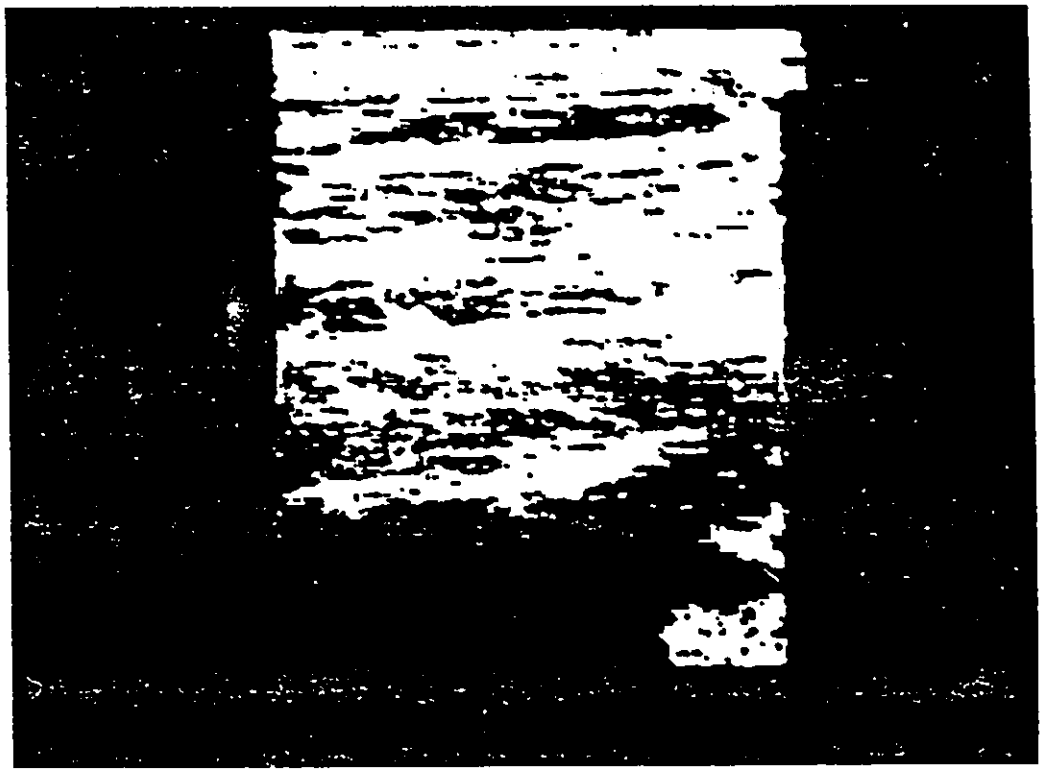

Figure 5.26: Région extraite après l'ajout de 10 régions

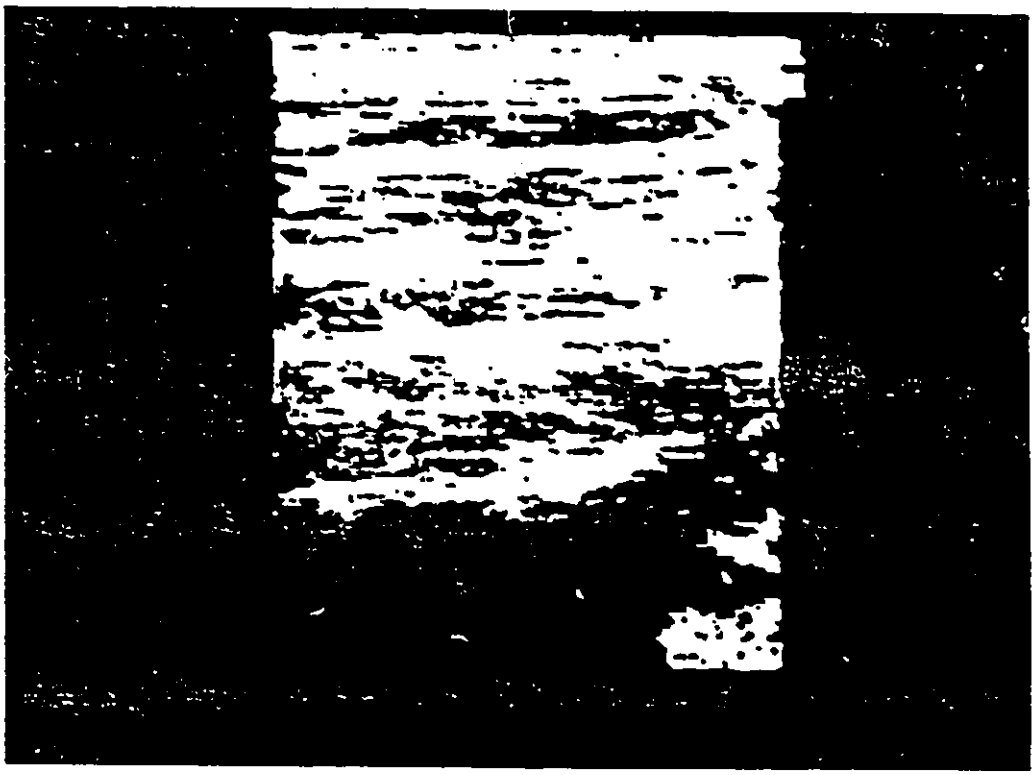

Figure 5.27: Région extraite après l'ajout de 20 régions

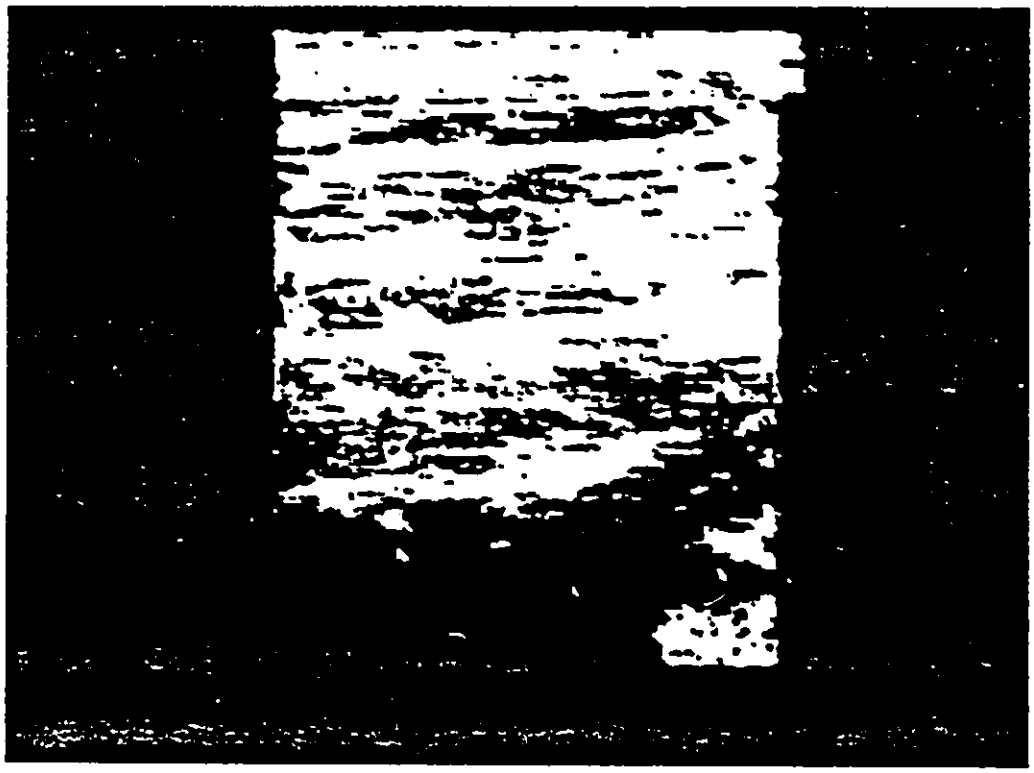

Figure 5.28: Région extraite après l'ajout de 30 régions

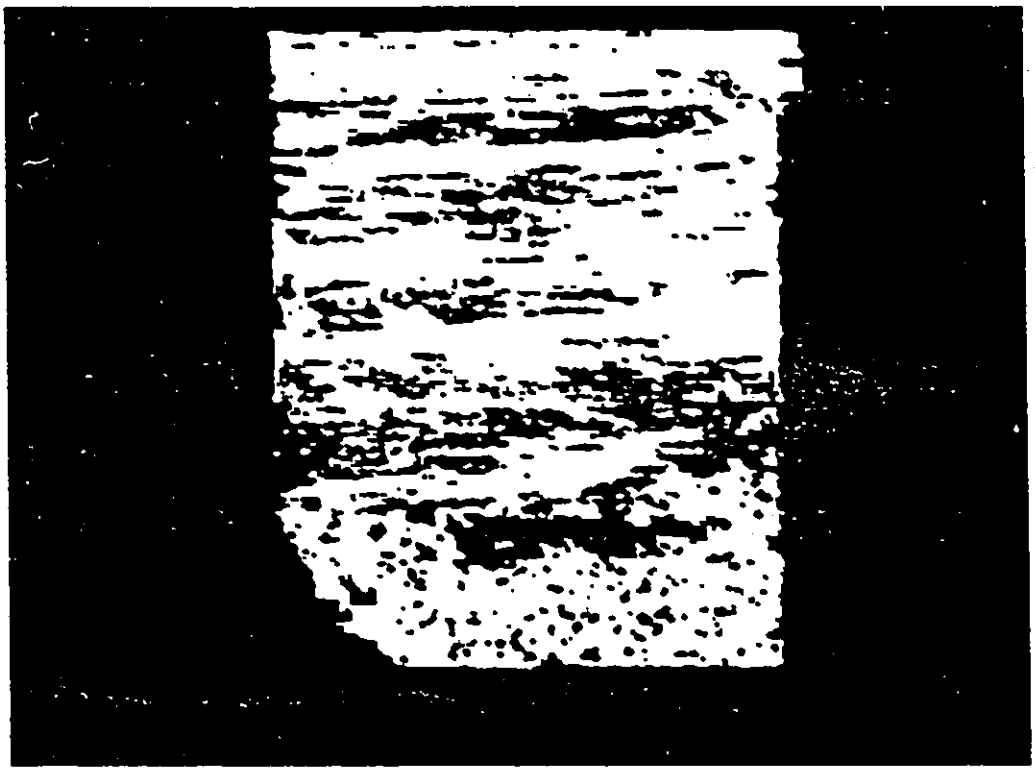

Figure 5.29: Région extraite après l'ajout de 39 régions

5.2.5 Résultat de la segmentation de la région de l'image ultrason

Comme nous l'avons mentionné précédemment, l'information de la région de l'image ultrason est contenue dans le niveau de gris de ses pixels. Il est donc important de s'assurer que la segmentation n'occasionne aucune perte importante de pixels appartenant à cette région. C'est pour cette raison qu'il est nécessaire d'évaluer le résultat de sa segmentation. Ceci est fait en comparant le résultat obtenu avec le résultat idéal. Nous calculons alors le nombre de pixels omis et le nombre de pixels ajoutés faussement à la région. Les résultats sont présentés au tableau 5.2.5.

Image	*Résultat obtenu*	*Résultat idéal*	*Erreur par omission*	*Ratio d'omission*	*Erreur par ajout*	*Ratio d'ajout*
im4-11.ima	188894	178114	1240	0.006962	12020	0.067485
im4-14.ima	157438	157438	0	0	0	0
im4-25.ima	198816	189505	0	0	9311	0.048133
im8-11.ima	156450	154735	0	0	1715	0.011083
global	701598	679792	1240	0.001824	23046	0.033902

Table 5.4: Résultats de la segmentation de la région de l'image ultrason

L'erreur par omission est plus sérieuse car elle signifie une perte d'information. Les pertes d'omission présentes pour l'image "im4-11.ima" ont été présentées au département d'ultrason de l'hôpital Civic d'Ottawa et ont été jugées acceptables, car le faible nombre de pixel manquant se situait le long des limites de l'image. Nous retrouvons aux figures 5.30 à 5.33 les limites des régions de l'échelle de gris et de l'image ultrason extraites pour les quatre images tests.

5.3 Segmentation de l'image de fond

L'extraction de cette région est triviale. Ses limites peuvent être déterminées par défaut en éliminant de l'image les deux premières régions extraites. Cependant l'information contenue

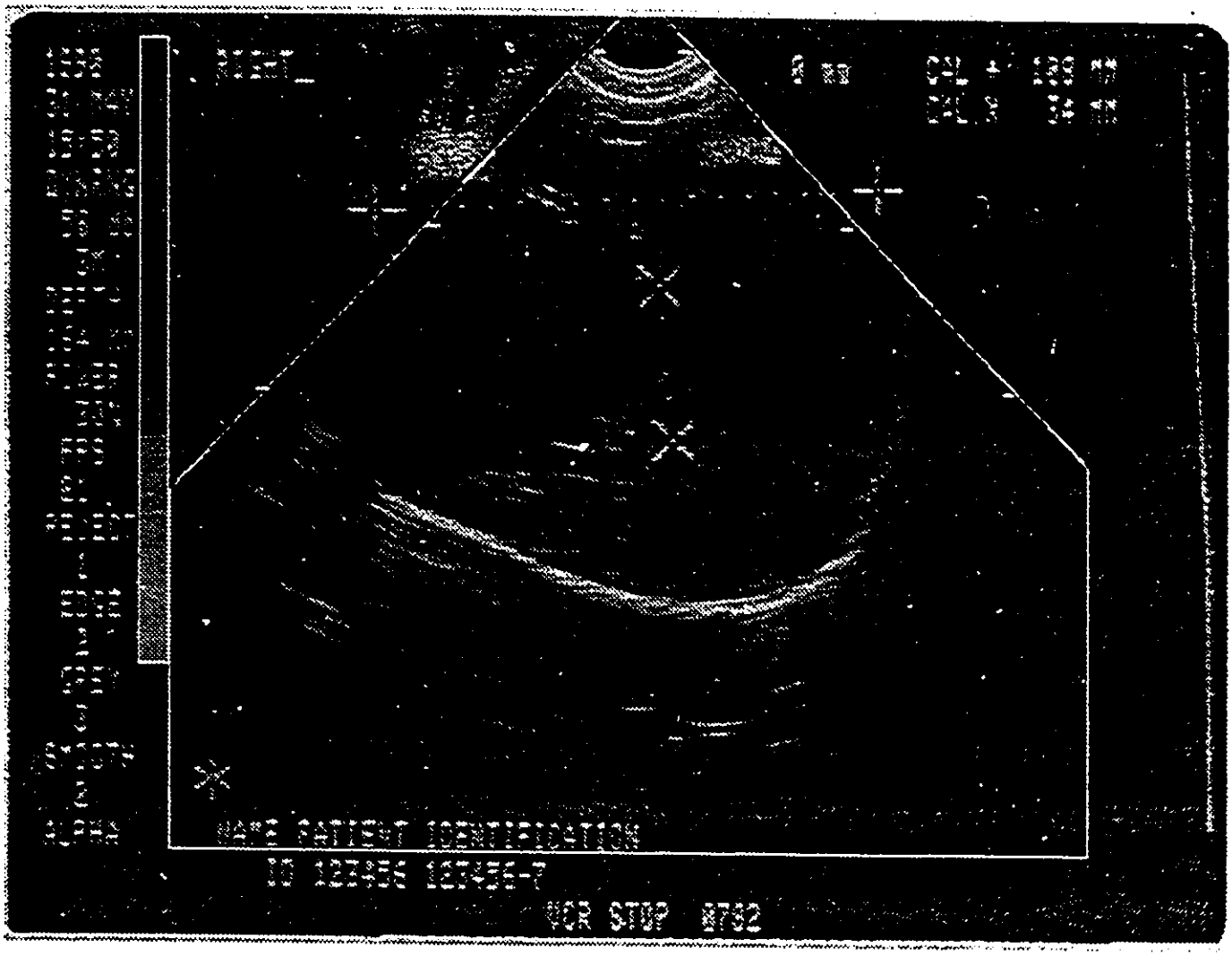

Figure 5.30: Limites pour l'image "im4-11.ima"

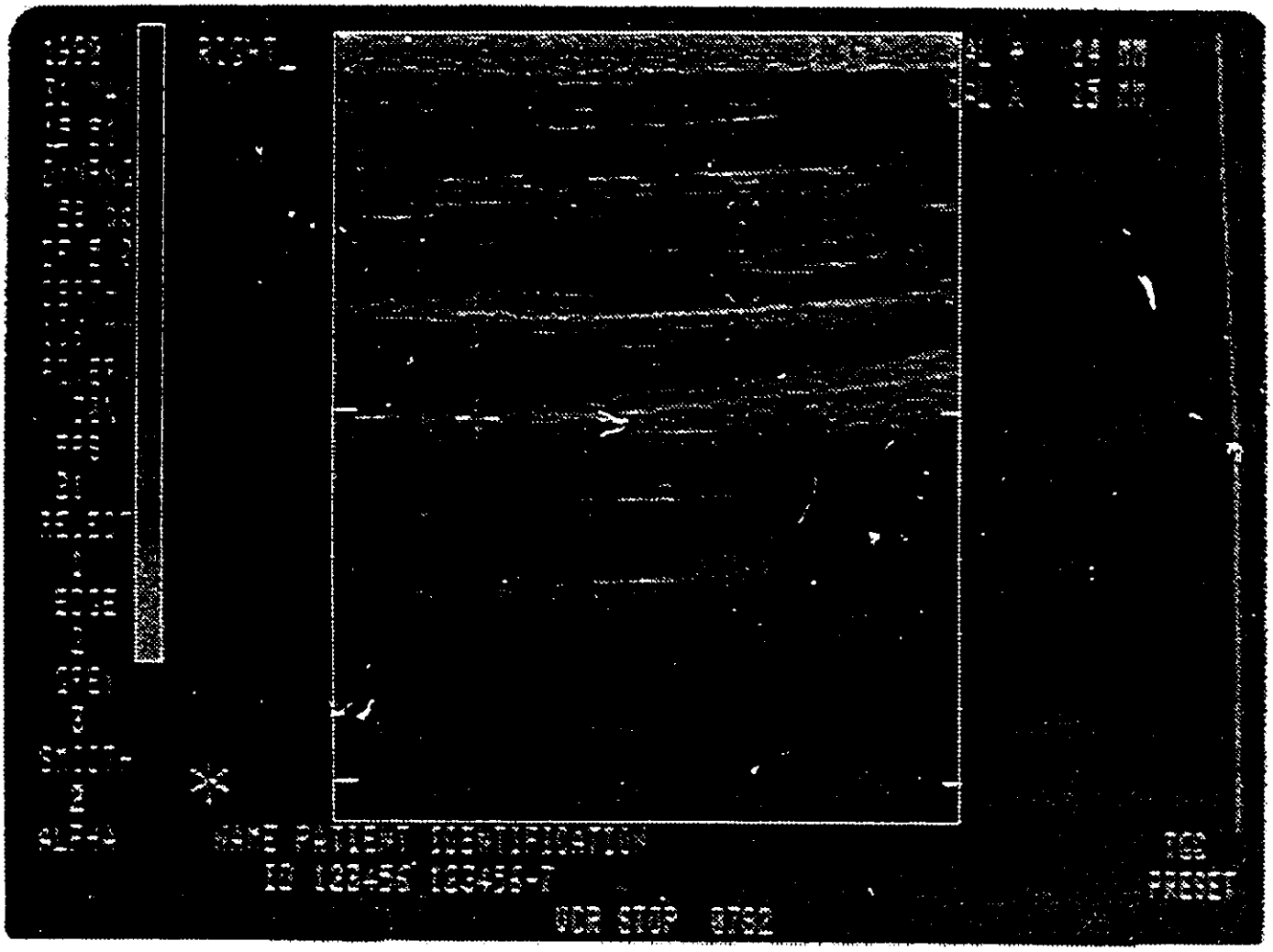

Figure 5.31: Limites pour l'image "im4-14.ima"

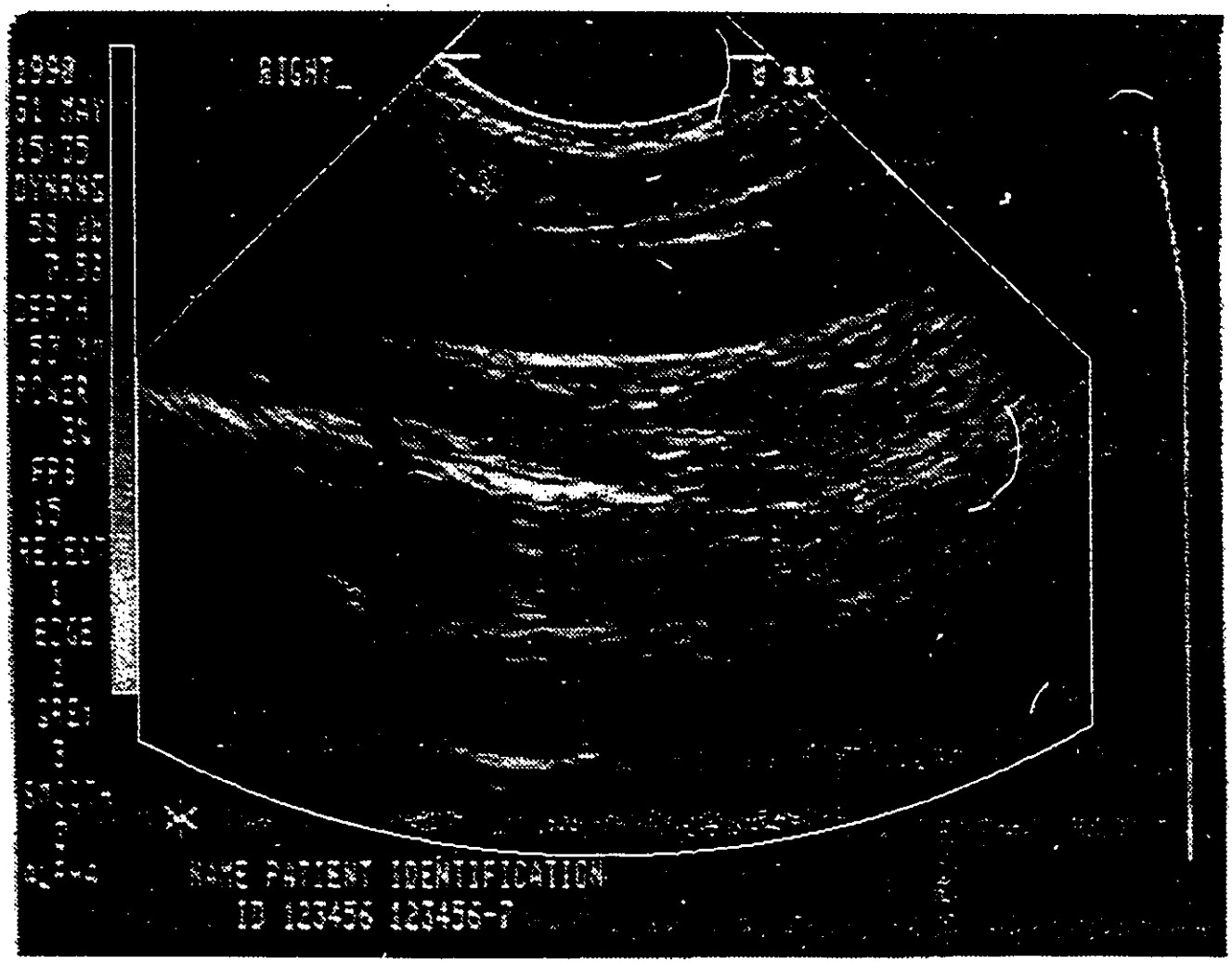

Figure 5.32: Limites pour l'image "im4-25.ima"

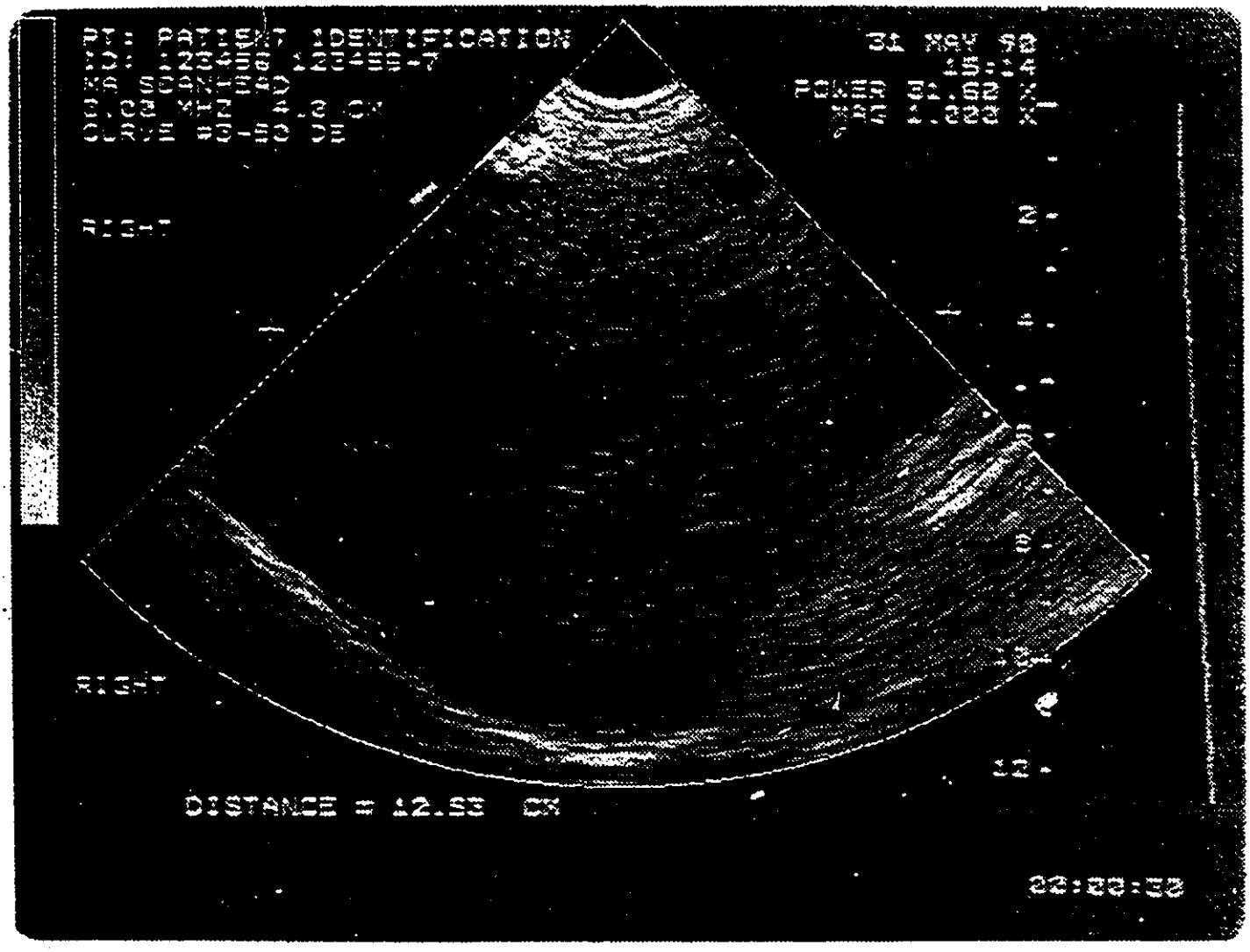

Figure 5.33: Limites pour l'image "im8-11.ima"

dans cette région ne nécessite pas la gamme complète des intensités présentes dans l'image originale. Il est possible de ramener l'information présente sous une forme plus simple. Au niveau du pixel elle se présente sous forme binaire. Un pixel est soit de forte intensité ou de faible intensité (blanc ou noir). Chaque pixel est donc classé selon ces deux groupes. Une façon facile de le faire est en établissant un seuil. Il existe plusieurs méthodes pour déterminer un seuil approprié. Une d'elle se base sur l'hypothèse que les pixels de fort gradient ont une intensité susceptible d'être utilisée comme valeur de seuil. L'opérateur développé par Marr et Hildreth [17] nous donne la dérivée seconde d'une version lissée de l'image par un filtre gaussien. De celle-ci nous extrayons les passages par zéro. Ceux-ci correspondent aux maximums locaux du gradient de l'image donc des points d'intensité susceptible d'être utilisée comme valeur de seuil. De plus les résultats pratiques nous montrent que ces points délimitent bien les régions de pixels de forte intensité. Il est donc raisonnable de croire que l'intensité maximale de ces points constituerait possiblement une bonne valeur de seuil.

Des essais avec un seuil fixé de manière à ce que 98% des pixels extraits par l'opérateur de Marr et Hildreth soient sous ce seuil nous ont donné des résultats de qualité non-uniforme à travers l'image. L'image "im4-11.ima" ainsi seuillée est présentée à la figure 5.34. Ceci est dû à la variation de l'intensité locale moyenne des pixels à travers l'image. L'utilisation d'un seuil variable à travers l'image est tout désignée pour corriger ce problème. L'image est donc divisée en 12 zones (voir figure 5.35) et un seuil différent est déterminé pour chacune d'elles. Les résultats obtenus à l'aide de cette méthode sont de beaucoup supérieurs. L'image "im4-11.ima" ainsi seuillée est présentée à la figure 5.36.

Une opération de reconnaissance de caractères peut suivre afin de regénérer ceux-ci sous une forme non bruitée. Ce sujet ayant fait l'objet de nombreuses recherches, nous ne nous y sommes pas attardés.

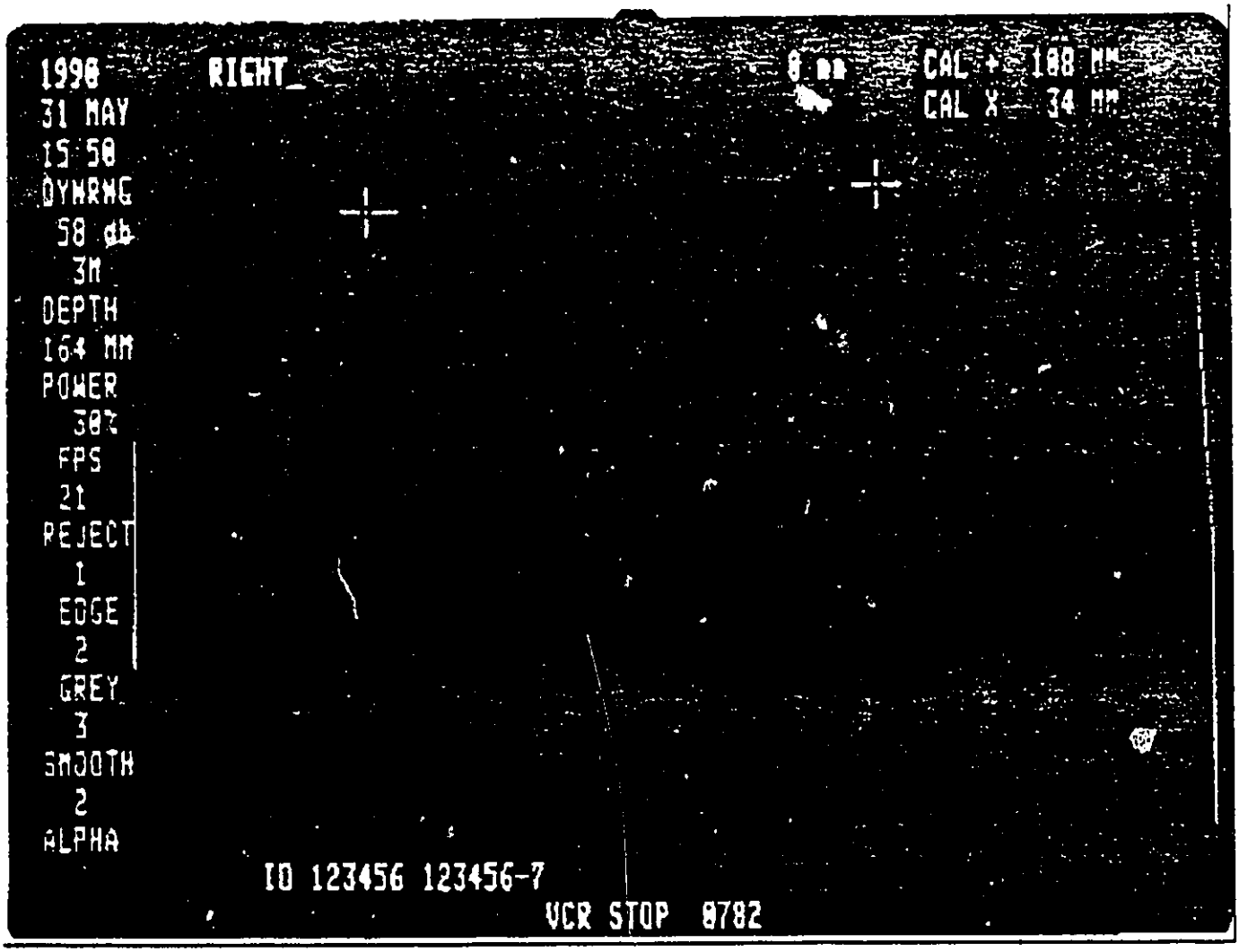

Figure 5.34: Image "im4-11.ima" seuillée à l'aide d'un seuil uniforme

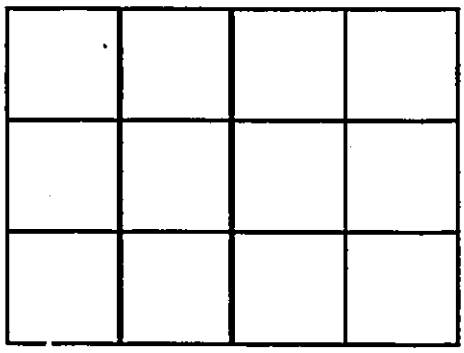

Figure 5.35: Division de l'image en 12 zones

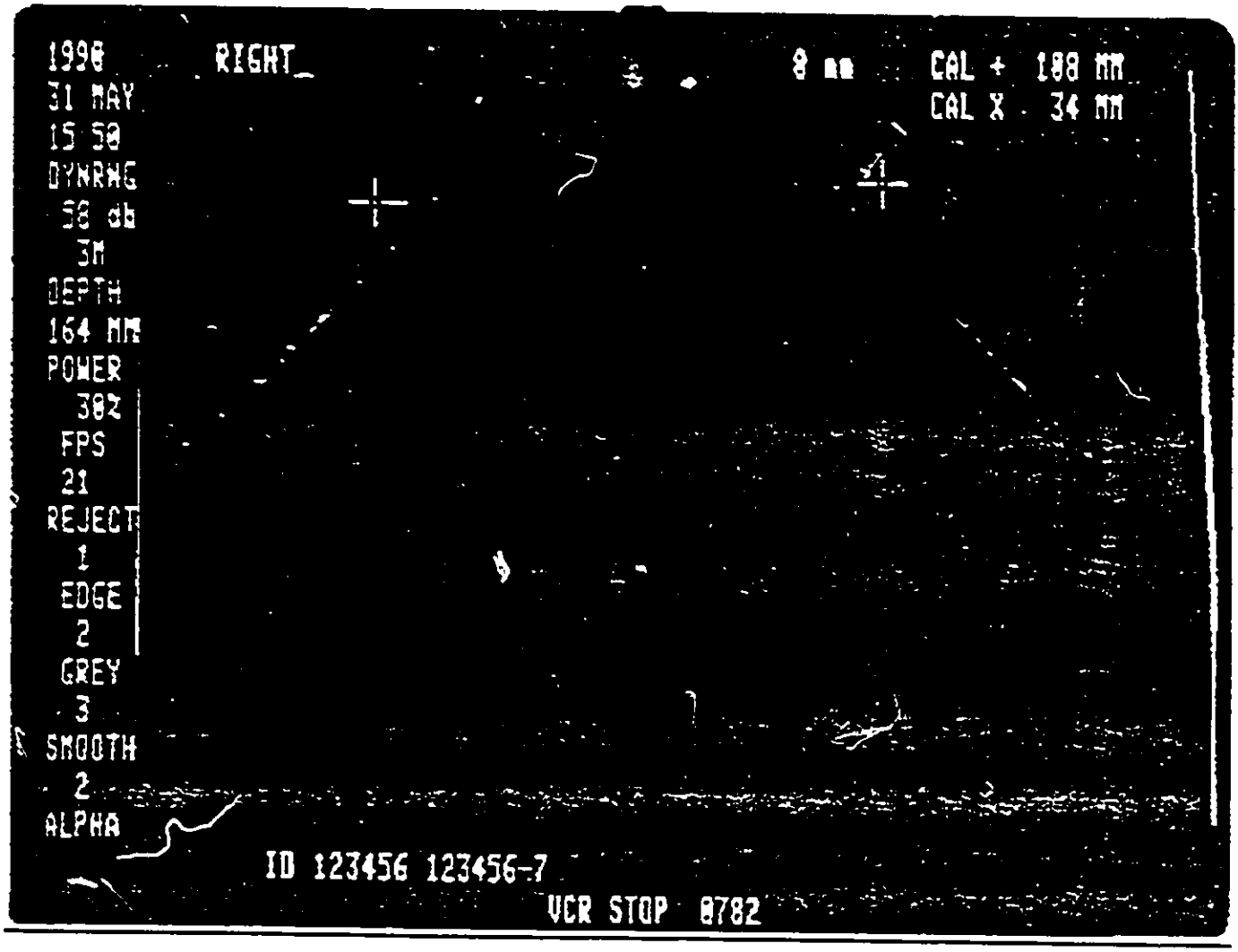

Figure 5.36: Image "im4-11.ima" seuillée à l'aide d'un seuil variable

Chapitre 6

Compression de l'information

Comme pour la segmentation, nous avons trois régions d'intérêt : "l'échelle de gris", "l'image ultrason", et "l'image de fond". Nous adresserons chacune de ces régions individuellement afin de choisir la technique qui exploite au maximum les caractéristiques de chacune de ces régions. Les images tests utilisées sont les mêmes que nous avons présentées au chapitre précédent soit les images : "im4-11.ima", "im4-14.ima", "im4-25.ima", et "im8-11.ima". Toutes ces images sont initialement codées à 8 bits/pixel. Cette valeur nous servira de point de référence pour l'évaluation du taux de compression.

6.1 Compression de l'échelle de gris

L'échelle de gris comme nous l'avons expliqué antérieurement est utilisée comme mesure de la qualité de développement du film. Un bon développement du film se traduira par une représentation sans saturation des hauts et bas niveaux de gris. La dynamique des niveaux de gris du moniteur y est représentée par échelon ou de manière continue. Chaque bloc pour le cas échelonné ou chaque ligne pour le cas intégral est d'une intensité constante. Il est donc inutile de coder tous les pixels mais plutôt la valeur de l'intensité de chaque bloc ou de chaque ligne. Pour simplifier le cas, nous utiliserons le modèle général de l'échelle intégrale. Le processus de numérisation de l'image occasionne cependant une détérioration de l'image. L'intensité des pixels d'une même ligne ne sera plus nécessairement constante. De plus l'orientation de

l'image peut avoir été modifiée de quelques degrés occasionnant une inclinaison des lignes composant l'échelle de gris. Les lignes de l'échelle de gris dans notre image numérisée peuvent alors être constituées de pixels appartenant aux lignes précédentes et suivantes. L'intensité de n'importe quel pixel sur une ligne ne peut donc être utilisée pour représenter l'intensité de la ligne. Nous choisissons donc pour celle-ci la valeur médiane des pixels d'une même ligne. Cette valeur est déterminée en ordonnant les valeurs d'intensité par ordre croissant et en choisissant la valeur située en position centrale. Cette méthode nous permet d'éliminer l'effet du bruit et du chevauchement des lignes.

Pour conserver l'information contenue dans l'échelle de gris il est nécessaire de coder l'intensité de chaque ligne ainsi que de la position de l'échelle de gris dans l'image et ses dimensions. Son stockage requiert donc 1 byte par ligne pour l'intensité, 4 bytes pour la position horizontale et verticale de l'échelle de gris et 4 bytes pour la dimension horizontale et verticale. Les résultats de la compression de l'échelle de gris pour nos quatre images tests sont présentés au tableau 6.1.

Image test	*Nombre de pixels avant compression*	*Nombre de pixels après compression*	*Bits/pixel*
im4-11.ima	6171	363	0.470588
im4-14.ima	5808	363	0.500000
im4-25.ima	5082	363	0.571429
im8-11.ima	6644	302	0.363636

Table 6.1: Résultats de la compression de l'échelle de gris

6.2 Compression de l'image ultrason

Comme nous l'avons vu précédemment deux caractéristiques des images peuvent être exploitées pour la compression de leur information. Il s'agit de la tolérance de l'oeil pour un niveau contrôlé de distortion de l'image et de la forte corrélation entre pixels adjacents.

Etant donné que nous désirons une compression sans perte d'information, seule la corrélation interpixel peut être utilisée. Sept différentes techniques de compression exploitant cette caractéristique ont été évaluées. Nous les décrirons d'abord rapidement pour ensuite comparer leur résultat.

6.2.1 Code de Huffman sur l'intensité des pixels

La première méthode exploite la non-uniformité de la distribution de la probabilité des intensités de gris des images. Un code à longueur variable permet d'exploiter cette propriété. Nous avons donc choisi de modéliser les données par l'intensité des pixels et de coder celles-ci à l'aide du code de Huffman. Un code fixe demeure efficace tant que la fonction de distribution de probabilité reste stationnaire. Ceci est rarement le cas pour les images. Il est donc nécessaire d'utiliser une modélisation des statistiques en deux étapes ou adaptatives afin d'adapter le code aux données à transmettre. Etant donné que les images ultrason font rarement partie d'une séquence d'images, l'adaptation du code d'une image à l'autre pourrait s'avérer très importante. Nous n'évaluerons donc que le cas de la modélisation en deux étapes. Nous devons donc tenir compte de l'espace mémoire occupé par le code lors du calcul du taux de compression. Le signal de l'image est constitué de 256 différents états pour chacun desquels nous assignons un mot de code. Nous avons donc 256 mots de code à transmettre. La longueur maximale de ces mots de code est de 256 bits. il est donc nécessaire de conserver 8 bits pour coder la longueur de chacun d'entre eux. Il est nécessaire de transmettre premièrement la longueur de chacun des mots de code afin de pouvoir les extraire du signal reçu. L'espace mémoire requis pour la transmission du code peut être calculé de la façon suivante :

$$\text{Espace mémoire occupé par le code} = 256 \times 8 \text{ bits} + \sum_{i=1}^{256} \text{longueur du mot de code i}$$

6.2.2 Code de Huffman sur le modèle markovien

La deuxième méthode fait appel à une modélisation markovienne de l'image. Dans ce modèle nous assumons que la probabilité d'obtenir une certaine intensité au pixel suivant ne dépend que de la valeur de l'intensité du présent pixel. Nous avons donc une fonction de distribution distincte pour chacune des intensités possibles. Pour chacune de ces fonctions de distribution il est possible de créer un code de Huffman. Si ces différents codes sont connus au codage et au décodage, le code approprié pour le pixel suivant sera choisi selon la valeur du présent pixel. Le problème de stationnarité de la fonction de distribution ne se pose pas ici car nous avons affaire à des fonctions de distribution conditionnelles qui demeurent relativement stables tant qu'une forte corrélation entre les pixels adjacents existe. Nous ne considérons donc que le cas du code fixe. Le code fut généré à l'aide de la distribution de probabilité obtenue avec les images tests "im4-11.ima" et "im4-25.ima". Le cas de la modélisation des statistiques en deux étapes ne présente aucun intérêt car il demanderait un espace mémoire considérable, de l'ordre de 900 000 bytes, pour la transmission du code uniquement.

Les autres codes évalués font également appel à la dépendance entre pixels adjacents mais la modélisation des données est différente. Ils se servent de prédicteurs comme opérateur de décorrélation du signal. Ils codent alors le signal de différence entre la valeur réelle de l'intensité du pixel suivant et la valeur de la prédiction. Les variations d'un code à l'autre résident dans le choix du prédicteur et dans la méthode d'encodage du signal de différence. Le schéma général de cette procédure est présenté à la figure 6.1.

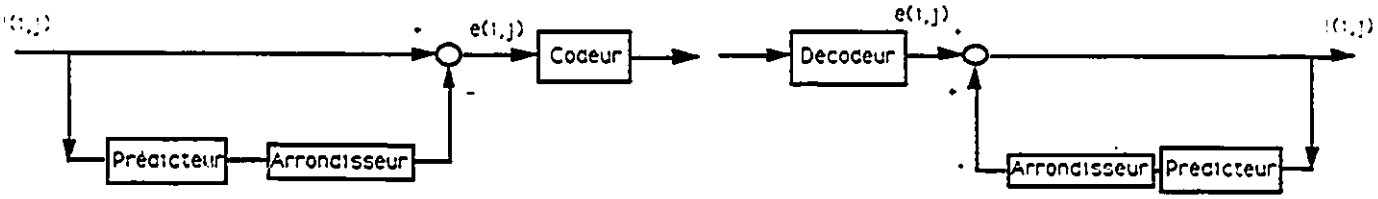

Figure 6.1: Schéma général de codage du signal de différence

Deux prédicteurs et quatre méthodes d'encodage ont été évalués. Nous avons d'abord

établi le meilleur prédicteur pour ensuite évaluer les méthodes de codage avec celui-ci.

Le premier prédicteur est un prédicteur d'ordre 0 qui consiste à prendre la valeur du pixel précédent de la même ligne comme prédiction. Le signal de différence est défini comme suit :

$$e_0(i,j) = I(i,j) - \hat{I}_0(i,j) \tag{6.1}$$
$$\text{où } \hat{I}_0(i,j) = I(i,j-1)$$

Le deuxième est un prédicteur d'ordre 1 qui consiste à prendre la moyenne du pixel précédent de la même ligne et du pixel correspondant à la même position de la ligne précédente. Le signal de différence est défini comme suit :

$$e_1(i,j) = I(i,j) - \hat{I}_1(i,j) \tag{6.2}$$

$$\text{où } \hat{I}_1(i,j) = \lfloor \frac{I(i,j-1) + I(i-1,j)}{2} \rfloor$$

La valeur de cette prédiction étant le résultat d'une division, il est possible que la valeur de cette prédiction ne soit pas un nombre entier. Il est donc nécessaire d'arrondir la valeur de cette prédiction à l'entier le plus près afin que le signal de différence soit un entier et que nous puissions obtenir la valeur originale à l'étape de reconstruction du signal. Ces deux prédicteurs ont été évalués avec le code de Huffman pré-transmis. Le prédicteur d'ordre 0 s'est avéré toujours supérieur (voir figure 6.2 à 6.6). Le prédicteur d'ordre 0 est donc celui que nous avons utilisé pour l'évaluation des méthodes d'encodage sur un signal de différence.

6.2.3 Code de Huffman sur la différence d'intensité

Examinons maintenant les méthodes d'encodage que nous avons évaluées sur ce signal de différence. La première, comme nous l'avons mentionné ci-dessus, est le code de Huffman. La fonction de distribution de probabilité du signal de différence est suffisamment stable pour qu'une modélisation fixe des statistiques nous donne de bons résultats. Ici également,

le code fut généré à l'aide de l'histogramme obtenu avec les images tests "im4-11.ima" et "im4-25.ima".

Un code adapté à la fonction de distribution de chaque image pourrait nous donner de meilleurs résultats. Dans ce cas il est nécessaire de transmettre le code préalablement au signal. Nous devons donc tenir compte de l'espace mémoire requis pour le code lors du calcul du ratio de compression. Le signal de différence est constitué de 512 différents états pour chacun desquels nous assignons un mot de code. Nous avons donc 512 mots de code à transmettre. La longueur maximale de ces mots de code est de 512 bits, il est donc nécessaire de conserver 9 bits pour coder la longueur de chacun d'entre eux. L'espace mémoire requis pour la transmission du code peut être calculé de la façon suivante :

$$\text{Espace mémoire occupé par le code} = 512 \times 9 \text{ bits} + \sum_{i=1}^{512} \text{longueur du mot de code i}$$

Ces deux alternatives ont donc été évaluées.

6.2.4 Code de changement de niveau sur 2 et 4 bits

Les deux prochaines méthodes d'encodage que nous avons évaluées sont ce que Redfern et Hines [24] appellent un "shift level code" (code de changement de niveau). Il s'agit comme nous l'avons expliqué au chapitre 3 d'un code dont les mots de code sont définis sur plusieurs niveaux. Pour que ce code soit efficace, il est nécessaire que le signal de différence soit concentré dans les faibles valeurs. Ce code peut être défini sur 2 ou 4 bits.

6.2.5 Code des différences communes

La dernière méthode d'encodage que nous avons évaluée est ce que Suzan [32] nomme "differences technique" (code des différences communes). Comme nous l'avons expliqué au chapitre 3 les quinze valeurs les plus probables sont codées sur 4 bits, et un mot de code est réservé pour représenter toutes les autres valeurs.

On retrouve aux figures 6.2, 6.3, 6.4, 6.5 et 6.6 les résultats obtenus avec chacune des méthodes de compression appliquées au signal de différence obtenu avec le prédicteur d'ordre 0 soit le "code de Huffman fixe et pré-transmis", le "code de changement de niveau sur 2 bits", le "code de changement de niveau sur 4 bits", et le "code des différences communes ". De plus, on y retrouve les résultats pour le "code de Huffman pré-transmis" appliqués au signal de différence obtenu avec le prédicteur d'ordre 1 et au signal original ainsi que les résultats pour le "code de Huffman fixe" pour le modèle markovien.

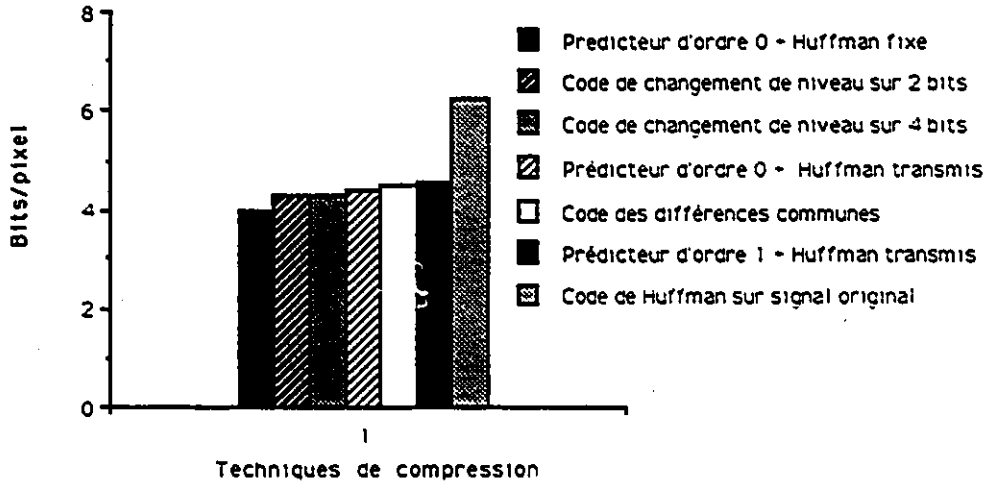

Figure 6.2: Résultats de compression sur l'image "im4-11.ima"

Le prédicteur d'ordre 0 associé au code de Huffman fixe nous donne pour toutes les images tests la meilleure compression. Nous obtenons une compression qui ne demande que 4.204440 bits/pixel sur l'ensemble de nos quatre images tests. Le "code de changement de niveau sur 4 bits" est également efficace avec 4.475446 bits/pixel. Ce sont cependant deux codes à longueur variable ce qui peut augmenter la complexité du décodage soit au niveau des délais ou de l'algorithme de reconnaissance du mot de code. Le "code des différences communes" est un code à longueur "semi-variable". Ses mots de code sont tous de même longueur dont l'un d'eux est complété par un code de 8 bits. Ses performances sont inférieures bien

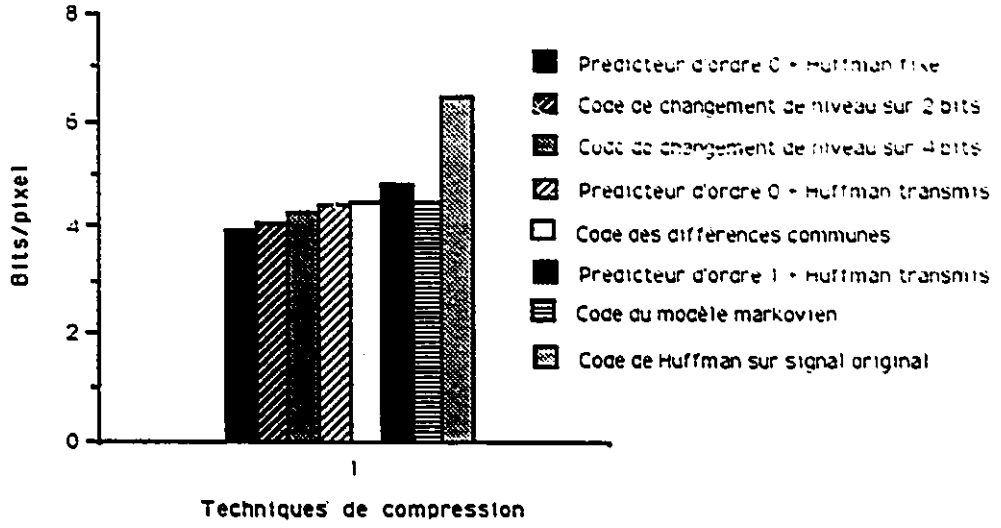

Figure 6.3: Résultats de compression sur l'image "im4-14.ima"

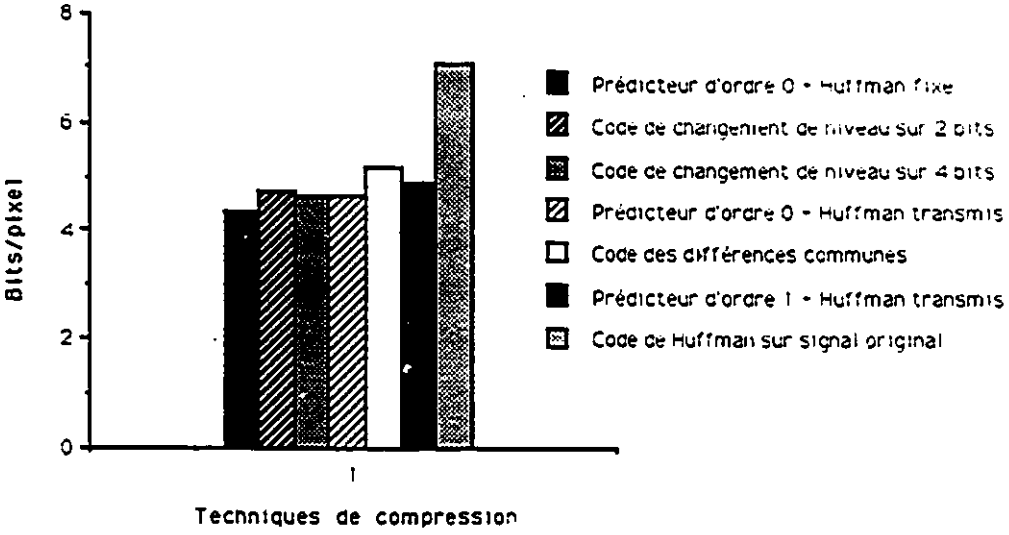

Figure 6.4: Résultats de compression sur l'image "im4-25.ima"

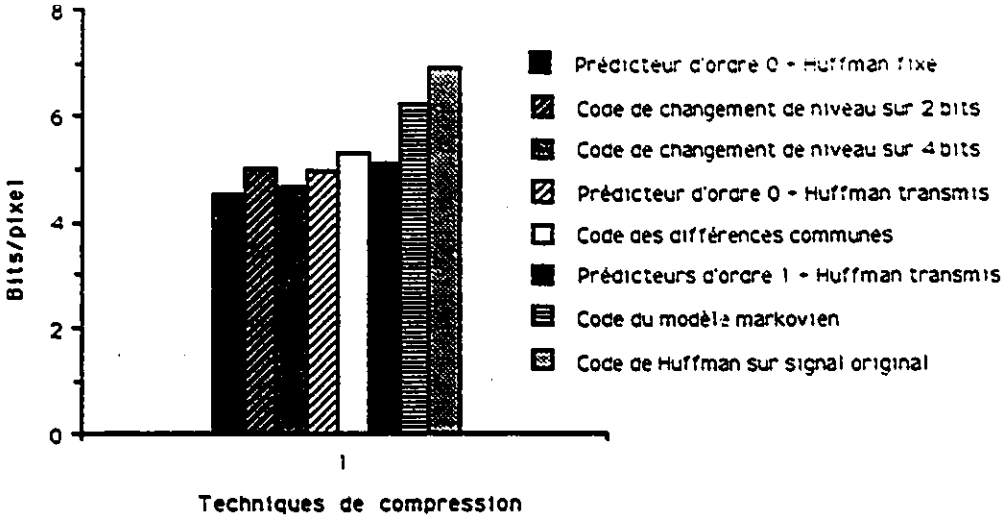

Figure 6.5: Résultats de compression sur l'image "im8-11.ima"

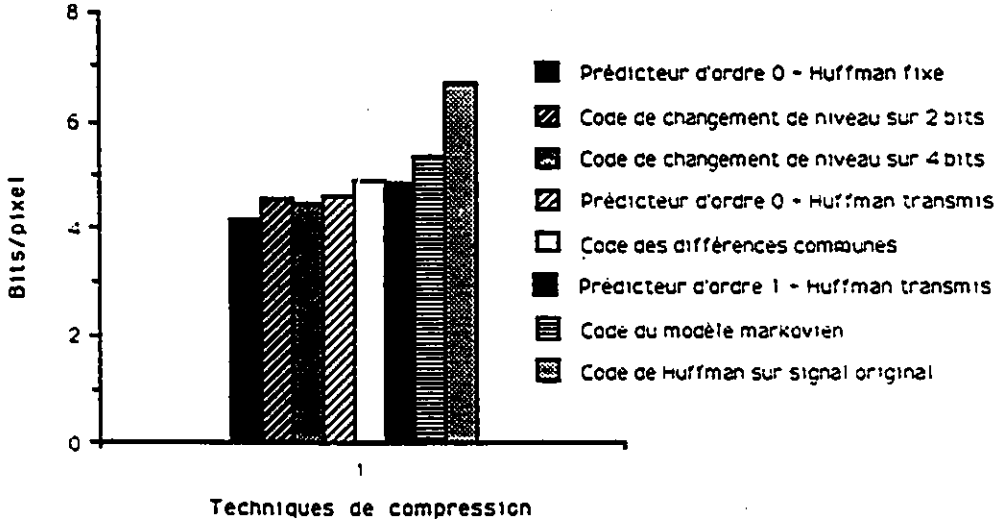

Figure 6.6: Résultats de compression moyenne sur les quatre images

qu'acceptables avec 4.868646 bits/pixel.

6.3 Compression de l'image de fond

L'image de fond comprend les caractères imprimés, les pointeurs et l'échelle de correction d'intensité. N'ayant pas poussé le prétraitement de cette région jusqu'à une opération de reconnaissance de caractère, toutes les informations seront considérées comme étant binaires. Le signal est donc composé de longs segments noirs et de courts segments blancs. Cette particularité du signal le prête bien à un codage de type "run-length". Celui-ci consiste à modéliser le signal par bloc de pixels d'intensité constante. Un bit est utilisé pour représenter l'intensité et un certain nombre de bits est utilisé pour représenter la longueur du bloc. Le nombre de bits requis pour coder cette dernière varie selon la longueur maximale de bloc permise. Le nombre de bits nécessaires pour chaque mot de code en fonction de la longueur maximale permise est présenté dans le tableau 6.3.

Longueur maximale des blocs (pixels)	*Nombre de bits*
8	4
16	5
32	6
64	7
128	8
256	9
512	10
1024	11
2048	12

Table 6.2: Nombre de bits nécessaires en fonction de la longueur maximale

En fait, l'intensité choisie pour représenter les pixels noirs n'a pas été fixée à "0" mais plutôt à l'intensité moyenne des pixels de cette région sous le seuil. Ce choix nous fut recommandé par le personnel médical du département d'ultrason de l'hôpital Civic d'Ottawa. Cette intensité produit une image se rapprochant des conditions habituelles de visionnement.

Le signal produit par le codage "run-length" présente des concentrations importantes de longs blocs de pixels noirs et de courts blocs de pixels blancs. Un tel signal se prête bien à une compression additionnelle à l'aide du code de Huffman. Ici le même dilemme, que dans le cas de "l'image ultrason", se pose à savoir s'il est préférable de transmettre un code de Huffman adapté à chaque image avant la transmission de celles-ci ou d'établir un code de Huffman, moins performant, commun à toutes les images. Ces trois méthodes soit le code "runlength", le code "runlength" suivi du code de Huffman pré-transmis et le code "runlength" suivi du code de Huffman fixe, ont été évaluées sur nos quatres images tests. Le code fixe fut généré à l'aide de la distribution de probabilité obtenue avec les images tests "im4-11.ima" et "im4-14.ima". Les résultats sont présentés aux figures 6.7 à 6.11.

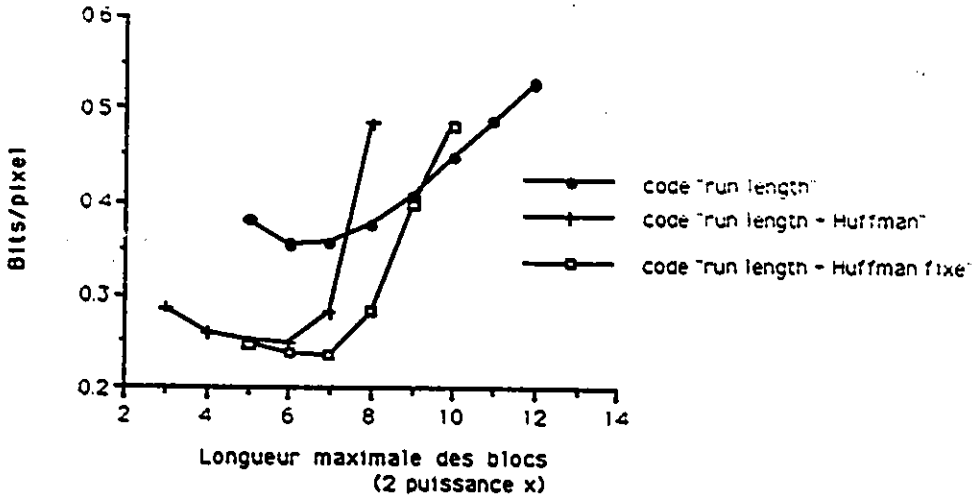

Figure 6.7: Compression de l'information binaire de l'image "im4-11.ima"

Le codage "runlength" pour des blocs de longueur maximale de 128 pixels suivi d'un "code de Huffman fixe" nous donne la meilleure compression soit 0.235017 bits/pixel pour les quatre images. Le codage "runlength" pour des blocs de longueur maximale de 64 pixels suivi d'un "code de Huffman transmis" nous donne un taux moyen de compression de 0.242311 bits/pixel. Les meilleurs résultats obtenus avec chaque méthode sur l'ensemble des quatre images sont comparés et présentés à la figure 6.12.

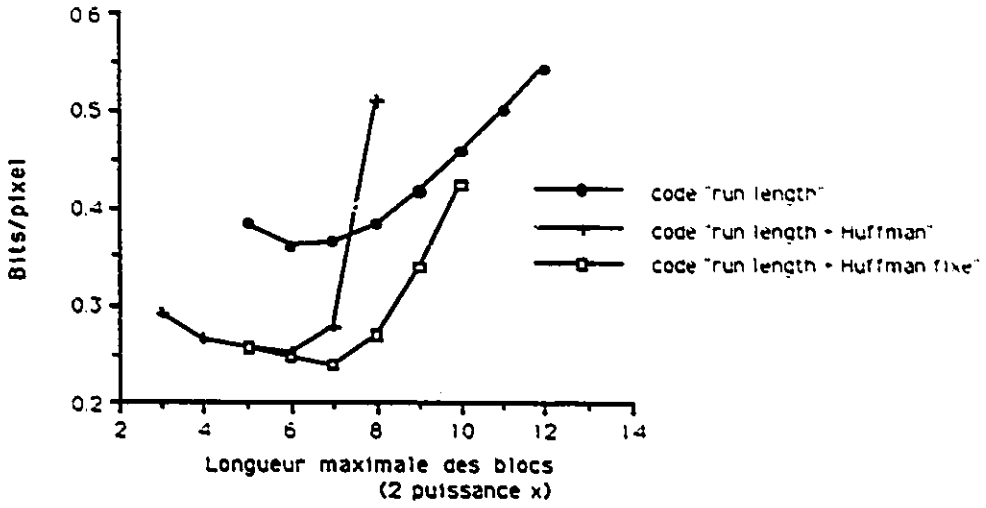

Figure 6.8: Compression de l'information binaire de l'image "im4-14.ima"

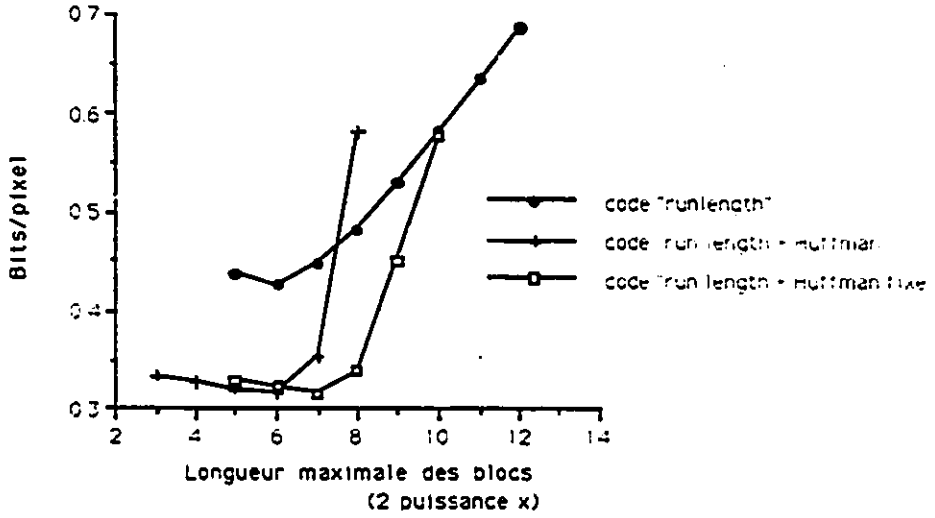

Figure 6.9: Compression de l'information binaire de l'image "im4-25.ima"

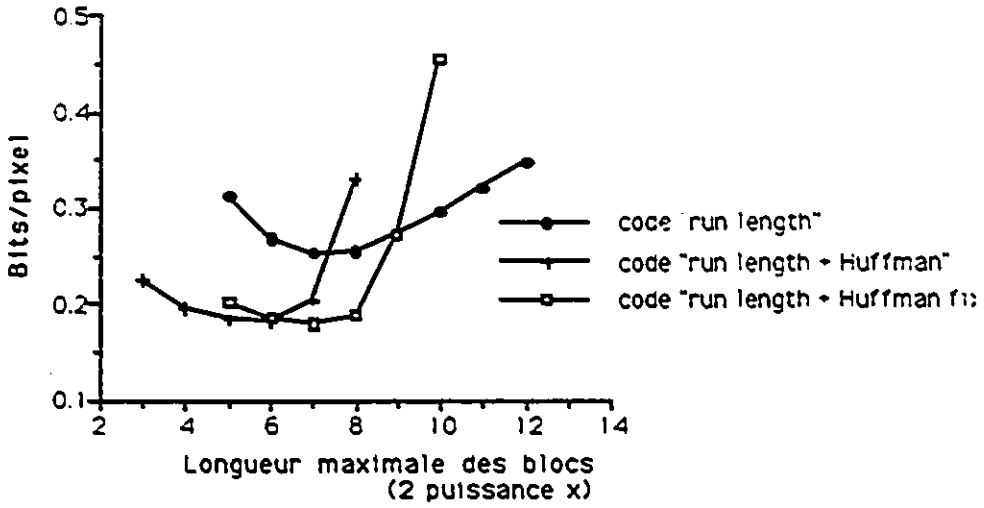

Figure 6.10: Compression de l'information binaire de l'image "im8-11.ima"

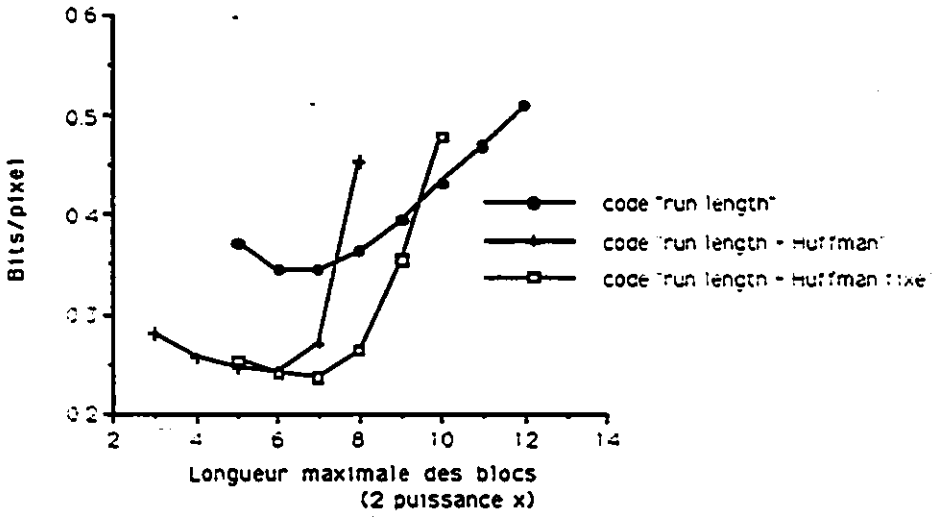

Figure 6.11: Compression de l'information binaire des quatre images

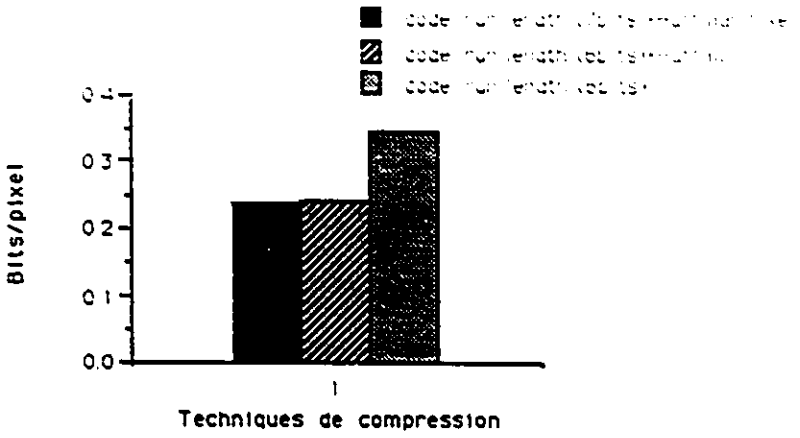

Figure 6.12: Meilleurs résultats de compression de l'information binaire

6.4 Compression globale de l'image

L'étape de la reconstruction de l'image demande un certain nombre d'informations générales sur notre image et sur les régions que nous avons extraites. Ces informations doivent donc être codées. L'espace mémoire occupé par celles-ci doit être comptabilisé lors de l'évaluation globale de la compression des images. Le tableau 6.4 présente ces différentes informations et l'espace mémoire requis pour leur codage.

Informations	*Espace mémoire (bytes)*
Dimensions de l'image	4
Forme de la région de l'image ultrason	2
Paramètres des limites de la région de l'image ultrason	72
Limites de la région de l'échelle de gris	8
Nombre de pixels dans chacune des régions	6
Total	92

Table 6.3: Informations générales sur l'image et ses régions

Un nombre fixe de 92 bytes est donc requis pour coder ces informations. Le résultat global de la compression des images ultrason est obtenu en additionnant ce nombre au nombre de

bits requis pour coder chacune des régions. Le nombre moyen de bits pour coder chacune des images et l'ensemble des quatre images est présenté aux figures 6.13 à 6.17 en fonction de la combinaison des méthodes employées pour coder chaque région.

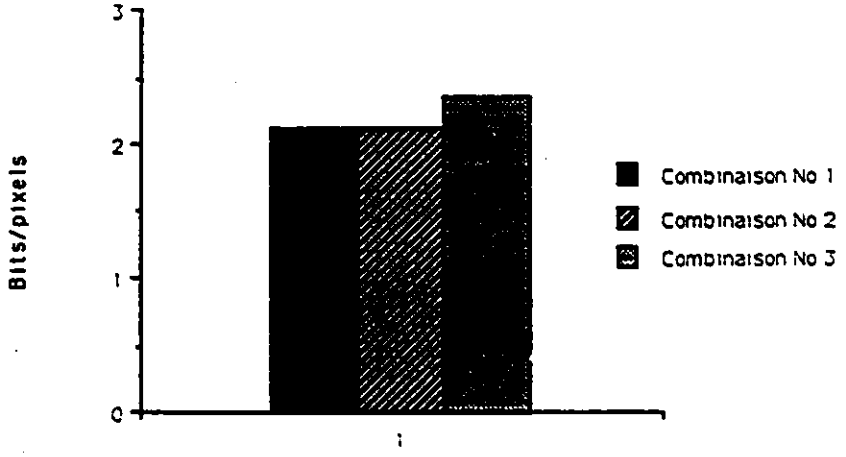

Figure 6.13: Compression globale de l'image ultrason "im4-11.ima"

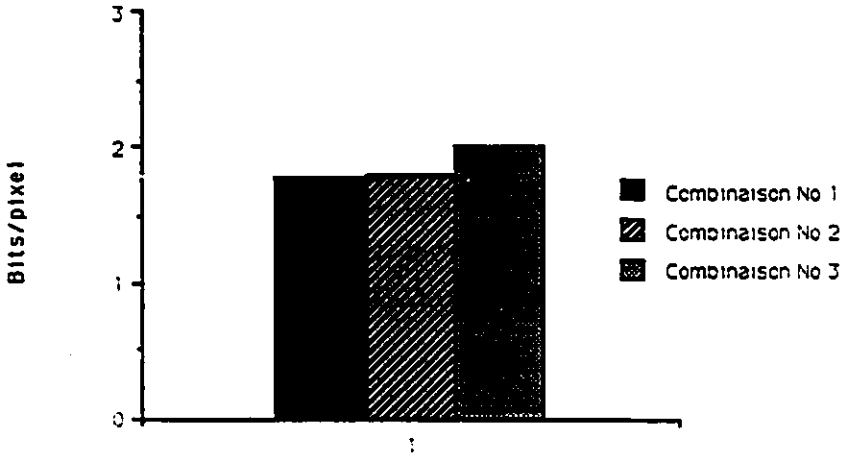

Figure 6.14: Compression globale de l'image ultrason "im4-14.ima"

Les méthodes considérées pour la compression de la région de "l'image ultrason" sont :

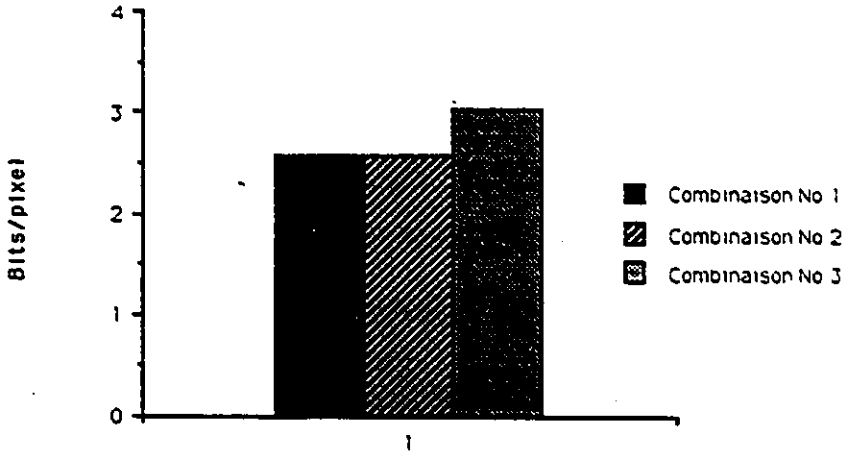

Figure 6.15: Compression globale de l'image ultrason "im4-25.ima"

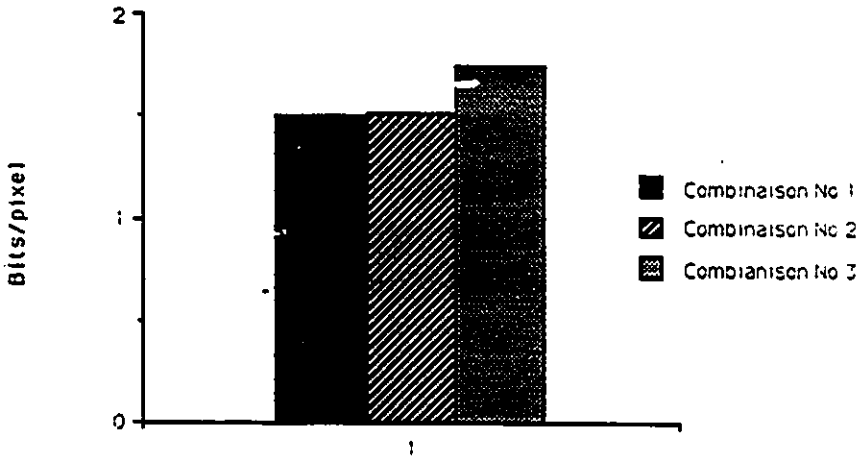

Figure 6.16: Compression globale de l'image ultrason "im8-11.ima"

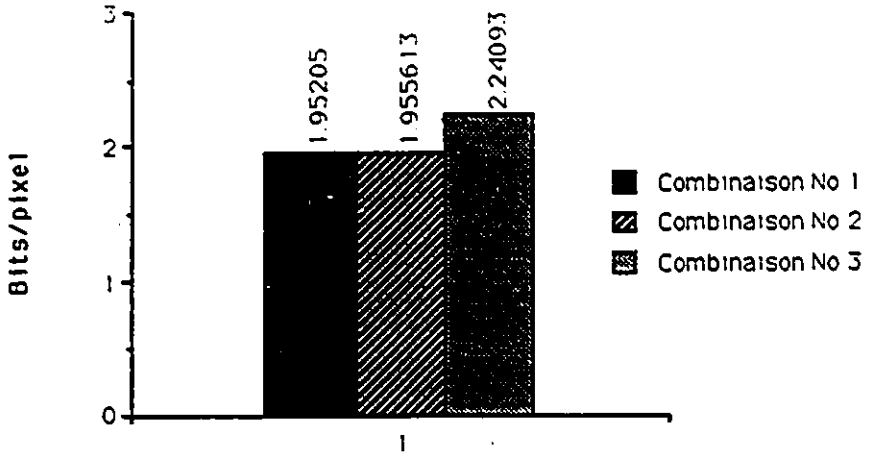

Figure 6.17: Compression globale moyenne des quatre images ultrason

1. Codage du signal de différence obtenu à l'aide du prédicteur d'ordre 0 par "code de Huffman fixe"

2. Codage du signal de différence obtenu à l'aide du prédicteur d'ordre 0 par " code de changement de niveau sur 4 bits"

3. Codage du signal de différence obtenu à l'aide du prédicteur d'ordre 0 par " code des différences communes"

Chacune de ces méthodes fut considérée en combinaison avec la méthode de codage "run-length" pour des blocs de longueur maximale de 128 pixels suivie d'un code de "Huffman fixe". Les résultats de la seule méthode de compression de la région de l'échelle de gris évaluée sont ajoutés à cette combinaison. La meilleure compression fut obtenue à l'aide de la combinaison No 1 avec 1.952050 bits/pixel soit un ratio de compression d'environ 4 : 1.

Le pourcentage de l'espace mémoire occupé par chacune des régions est présenté à la figure 6.18.

Les images obtenues après décompression de l'information sont présentées aux figures 6.19

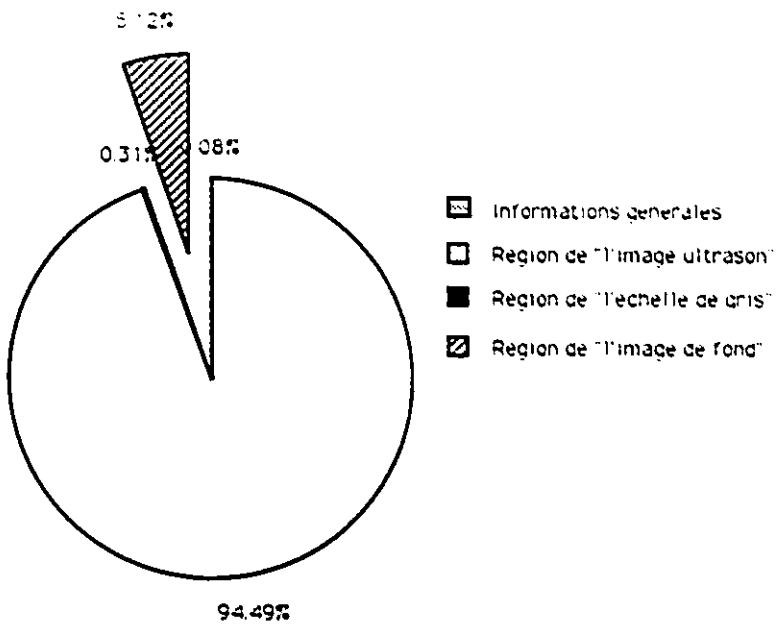

Figure 6.18: Pourcentage de l'espace mémoire occuppé par chacune des régions

à 6.22. Toutes ces images ont subi une perte d'information au sens strict du terme car elles comportent des distortions. Cependant il est important de revenir à la définition que l'on s'est faite de l'information contenue dans ces images. L'information se définit différemment pour chaque région et une distortion du signal ne signifie pas nécessairement une perte d'information. En effet pour la région de l'échelle de gris l'information consiste en la dynamique présente dans l'image après développement du film, pour la région de "l'image ultrason" l'information est contenue dans l'intensité de chaque pixel, et dans la région de "l'image de fond" l'information consiste en les caractères imprimés, les pointeurs et l'échelle de correction d'intensité , ou plutôt dans l'interprétation que l'on en fait. Ces définitions de l'information nous proposent un regard différent sur ce que l'on entend par perte d'information dans ces images. L'image "im4-11.ima" a subi une perte d'information dans sa région de "l'image ultrason" due à la segmentation trop sévère de celle-ci comme on peut le constater à la figure 6.19. Toutes les images ont subi une certaine perte d'information dans leur région de "l'image de fond" due à une certaine détérioration des caractères et des pointeurs. Cette détérioration des caractères, comme nous l'avons mentionné au chapitre précédent, peut être

corrigée par une opération de reconnaissance de caractères.

Perte d'information

dans la région de l'image ultrason

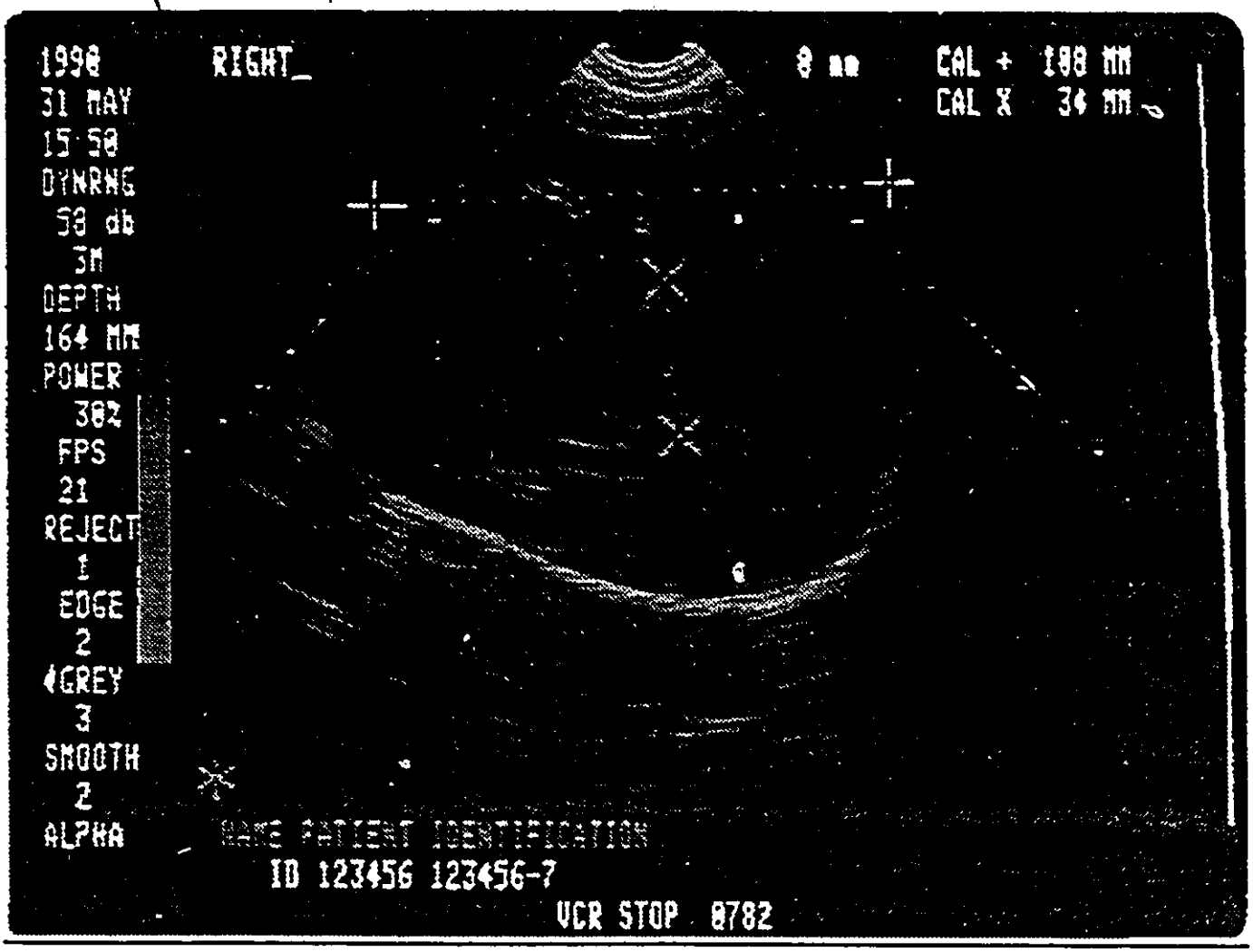

Figure 6.19: Image "im4-11.ima" reconstruite

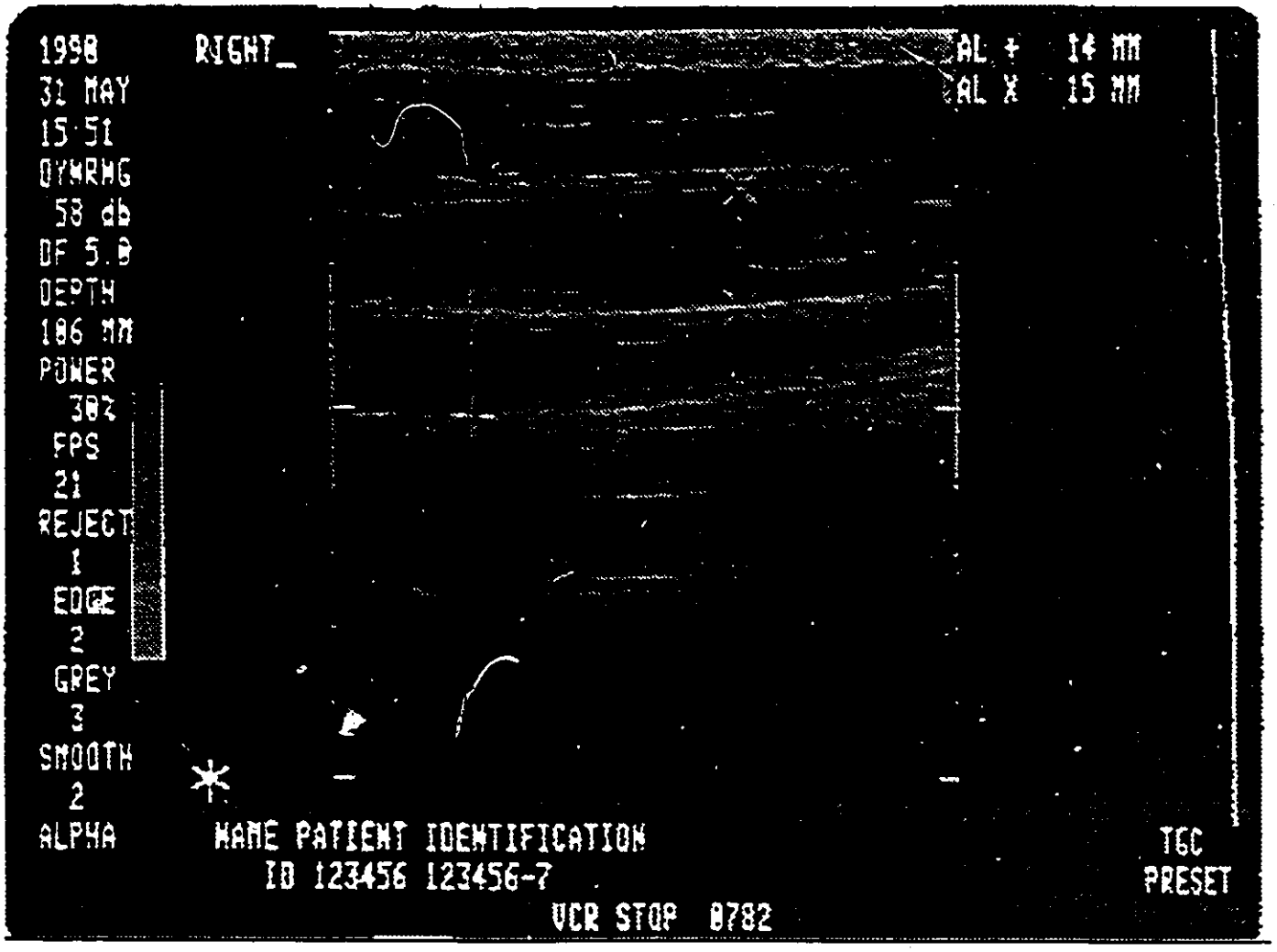

Figure 6.20: Image "im4-14.ima" reconstruite

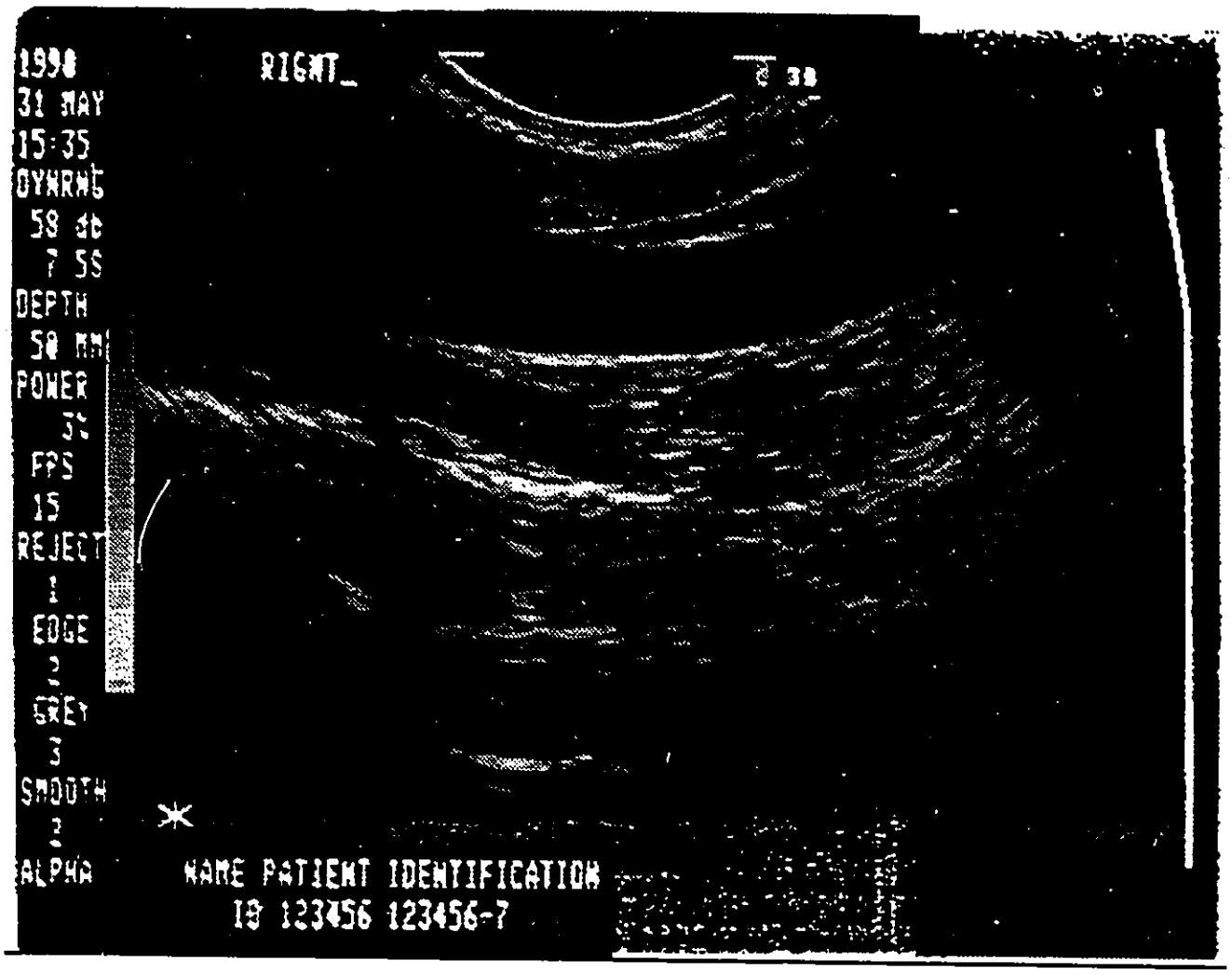

Figure 6.21: Image "im4-25.ima" reconstruite

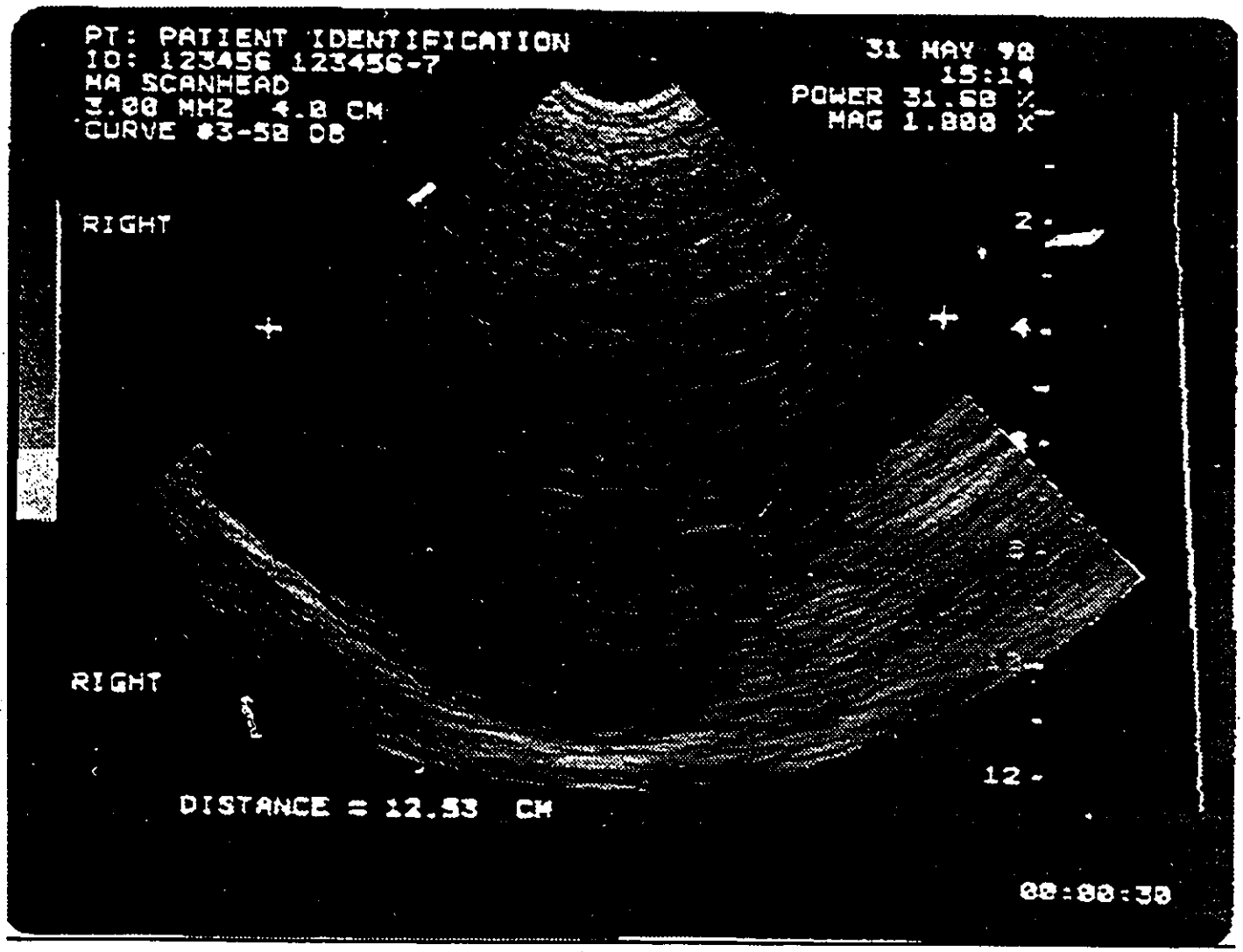

Figure 6.22: Image "im8-11.ima" reconstruite

Chapitre 7

Conclusions et suggestions

Nous avons développé un algorithme de compression des images ultrason sans perte significative d'information. Celui-ci utilise l'idée générale des méthodes de seconde génération soit une approche de l'information contenue dans l'image, sous sa forme structurale.

L'algorithme extrait tout d'abord les trois régions différentes des images ultrason soit l'échelle de gris, l'image ultrason, et l'image de fond. Différentes techniques de compression ont été évaluées sur chacunes de ces régions. Nous avons retenu comme méthode de compression pour la région de l'image ultrason le codage du signal de différence par code de "Huffman" fixe. Le signal de différence ayant été précédemment obtenu à l'aide d'un prédicteur d'ordre 0 utilisant le pixel précédent de la ligne comme prédiction. Nous avons retenu comme méthode de compression pour la région de l'image de fond un codage "runlength" de blocs de longueur maximale de 128 pixels suivi d'un code de "Huffman" fixe. Le signal pour cette région ayant été précédemment binarisé. La région de l'échelle de gris est codée en ne conservant que l'intensité de chaque ligne la constituant. Cette algorithme nous permet d'atteindre un taux de compression de l'ordre de 4 pour 1.

Certaines améliorations de l'algorithme pourraient être effectuées. Entre autre, le temps d'exécution pourrait être réduit de beaucoup en limitant les opérations de segmentation de l'image à une image de résolution réduite et en extrapolant les résultats sur toute l'image. Une procédure de reconnaissance des caractères pourrait êtra implantée ce qui augmenterait la

compression tout en donnant un signal de meilleure qualité. Finalement, il serait intéressant d'évaluer les performances des codes arithmétiques sur la région de l'image ultrason.

Références

[1] I. Bloch and F. Préteux, "Démarche morpho-mathématique pour la segmentation application aux lésions cérébrale en IRM", *Onzième colloque GRETSI*, Nice, 1-5 Juin 1987.

[2] J.F. Canny, "Finding edges and lines in images", *M.S. Thesis, MIT internal report AI-TR-720*. Juin 1983.

[3] D. Cheng and M. Goldberg, "An Algorithm for segmenting chest radiographs", *SPIE Visual Communications and Image Processing*, vol. 1001, pp. 261-268, Cambridge, Mass., novembre 1988.

[4] C.K. Chow and T. Kaneko, "Boundary detection of radiographic images by thresholding method", *in Frontiers of Pattern Recognition*, Academic Press, New York, 1972.

[5] L.S. Davis, "A survey of Edge Detection Techniques", *Computer Graphics and Image Processing*, vol. 4, pp.248-270, 1975.

[6] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, New York, 1972.

[7] R.C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley Publishing Company, 2ème édition, Don Mills, 1987.

[8] A.D. Gross and A. Rosenfeld, "Multiresolution Object Detection and Delineation", *Computer Vision, Graphics, and Image Processing*, vol. 39, pp. 102-115, 1987.

[9] R.M. Haralick, "Statistical and Structural Approaches to Texture", *Proceeding of the IEEE*, vol. 67, no. 5, pp.786-804, 1979.

[10] R.M. Haralick and L.G. Shapiro, "Survey: Image Segmentation Techniques", *Computer Vision, Graphics, and Image Processing*, vol. 29, pp. 100-132, 1985.

[11] D.R. Helman et G.G. Langdon, Jr "Data compression", *IEEE Potentials*, pp. 25-28, février 1988.

[12] M. Imme, "A New Method for Edge Detection", *Third International Conference on Image Processing and its Applications*, IEE, Université de Warwick, U.K., Juillet 18-20 1989.

[13] M. Kunt, "Progress in High Compression Image Coding", *International Journal on Pattern Recognition and Artificial Intelligence*, vol. 2, no 3, pp. 387-405, septembre 1988.

[14] M. Kunt, M. Benard, and R. Leonardi, "Recent Results in High-Compression Image Coding", *IEEE Transactions on Circuits and Systems*, vol. CAS-34, no 11, pp. 1306-1336, novembre 1987.

[15] H. Maître, "La segmentation des images", *Laboratoire Image, Département Images, Sons et Vidéo*, E.N.S.T., Paris, France, septembre 1989.

[16] M. Mansuripur, *Introduction to Information Theory*, Prentice-Hall Inc., 1987.

[17] D. Marr and E. Hildreth, "Theory of Edge Detection", *Proceeaings of the Royal Society of London*, B 207 pp. 187-217, 1980.

[18] G. Matheson, *Elements pour une théorie des milieux poreux*, Masson, Paris, 1967.

[19] O. Monga, "Segmentation d'images : Où en sommes nous ?", *Support de cours pour le Congrès PIXIM 89*, avril 1989.

[20] A.N. Netravali, "Picture Coding : A Review", *Proceeding of the IEEE*, vol. 68, no. 3, pp. 366-406, mars 1980.

[21] R. Ohlander, "Analysis of Natural Scenes", *PhD dissertion*, Université Carnegie-Mellon, Pittsburgh, Pa., 1975.

[22] J. Oswald, *Théorie de l'information ou Analyse diacritique des systèmes*, Masson, pp.89-114, Paris, 1986.

[23] D.P. Panda and A. Rosenfeld, "Image Segmentation by pixel classification in space", *IEEE Transaction on Computer*, C-27, pp. 875-879, 1978.

[24] A.J. Redfern and E.L. Hines, "Medical Image Data Compression Techniques", *Third International Conference on Image Processing and its applications*, IEE, Juillet 18-20 1989, Université de Warwick, U.K..

[25] E.M. Riseman and M. A. Arbib, "SURVEY: Computational Techniques in the Visual Segmentation of Static Scenes", *Computer Graphics and Image Processing*, vol. 6, pp. 221-276, 1977.

[26] P. Roos, M.A. Viergever, M.C.A. Van Dijke, and J.H. Peters, "Reversible Intraframe Compression of Medical Images", *IEEE Transactions on Medical Imaging*, vol. 7, no. 4, pp. 328-336, décembre 1988.

[27] A. Rosenfeld and A.C. Kak, *Digital Picture Processing*, vol. 2, Academic Press Inc., 2ème édition, New York, 1982.

[28] "Detection and Delineation of Compact Objects using Intensity Pyramids", *Pattern Recognition*, vol. 21, no 2, pp. 147-151, 1988.

[29] C.A. Sher and A. Rosenfeld, "Detecting and Extracting Compact Textured Regions using Pyramids", *Image and Vision Computing*, vol. 7, no 2, pp. 129-134, mai 1989.

[30] K.K. Shung, "General Engineering Principles in Diagnostic Ultrasound", *IEEE Engineering in Medicine and Biology Magazine*, pp. 7-13, décembre 1987.

[31] M. Suk and Chung, "A New Image Segmentation Technique Based on Partition Mode Test", *Pattern Recognition*, vol. 16, no 5, pp. 469-480, 1983

[32] S. Suzan, "Loss-Less Image Compression Techniques", *The Music Project Technical Report* Université d'Ottawa, Juillet 1989.

[33] J. Taboury and J.P. Monnier, *Guide pratique d'échographie abdominale*, Masson, 2ème édition, Paris, 1982.

[34] F. van der Heyden, "Evaluation of Edge Detection Algorithms", *Third International Conference on Image Processing and its Applications*, IEE, Université de Warwick, U.K., Juillet 18-20 1989.

[35] S. Watanabe and the CYBEST group, "An Automated apparatus for cancer prescreening: CYBEST", *Computer Graphics Image Process*, pp. 350-358, 1974.

[36] J.S. Weszka and A. Rosenfeld, "Threshold evaluation techniques", *IEEE Transaction on Systems Man and Cybernetic*, SMC-8, pp. 622-629, 1978.

[37] R. Wilson and M. Spann, *Image Segmentation and Uncertainty*, Research Studies Press Ltd., Letchworth, 1988.