

SEGMENTATION ADAPTATIVE POUR LE CODAGE D'IMAGES

THESE No 691 (1987)

PRESENTÉE AU DEPARTEMENT D'ELECTRICITE

ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ES SCIENCES TECHNIQUES

PAR

RICCARDO LEONARDI

ingénieur électricien diplômé EPFL
de nationalité italienne

acceptée sur proposition du jury :

Prof. M. Kunt, rapporteur
Prof. G. Coray, corapporteur
Prof. M. Eden, corapporteur
Prof. H. Maitre, corapporteur
Prof. F. Rocca, corapporteur

Lausanne, EPFL
1987

À mes parents

Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune
Or to take arms against a sea of troubles,
And by opposing, end them.

Hamlet (III,1), Shakespeare.

Remerciements

Je tiens à remercier tout d'abord mon directeur de thèse, le Professeur Murat Kunt. Grâce à lui, j'ai appris à mener à bien un travail de recherche. Pendant trois ans et demi, nous avons étroitement collaboré. Il a su me démontrer l'importance d'organiser son propre travail, le poids qu'il faut mettre dans la présentation de ce dernier pour qu'il puisse être apprécié par d'autres. Par ses nombreux contacts internationaux, j'ai pu connaître la communauté scientifique du traitement du signal, qui je l'espère me sera fort utile tout au long de ma carrière professionnelle. Enfin, je remercie un homme qui restera pour moi un ami.

Je remercie également mes collègues de bureau Michel Bénard et Patrick Volet dont les conseils m'ont toujours été fort utiles. Leur manière très diverse de voir les problèmes n'a pu qu'amener de l'eau à mon moulin.

Mes remerciements vont ensuite à tous ceux qui ont relu le manuscrit de ma thèse, et dont les nombreux commentaires m'ont été fort précieux. En l'occurrence, il s'agit du Professeur Murray Eden et des Docteurs Martin Vetterli et Roberto Cusani. Je leur sais gré du plaisir que j'ai eu à collaborer avec eux et j'espère que nous pourrons continuer à nous aider réciproquement dans nos activités de recherche.

Je remercie aussi les membres de mon jury d'avoir accepté de participer à l'évaluation de ce travail et pour les commentaires utiles qu'ils ont pu m'apporter.

Merci également au Professeur Frédéric de Coulon, Directeur du Laboratoire de Traitement des Signaux pour son regard critique face aux problèmes étudiés (quantification vectorielle).

Je tiens à remercier aussi le Professeur Henri Nussbaumer, qui m'a permis d'accéder aux moyens informatiques de son laboratoire pour la rédaction de ce travail.

Un merci tout particulier à Michel Kocher, qui ayant posé les jalons de la philosophie de ce travail m'a permis d'arriver aux résultats finaux. Souvent, ses conseils et son expérience se sont avérés d'une très grande utilité.

J'adresse enfin mes chaleureux remerciements à tous les membres actuels et passés du Laboratoire de Traitement des Signaux pour leur aide, leurs conseils et leur soutien; en particulier, Michael Unser, Anastasios Ikononopoulos, Pierre-Alain Moret, Jean-Charles Pandazis, Pierre-François Regamey, Erwin Beck, Roger Vuistiner, Jean-Marc Vesin, Mario Blais, Eric Gruaz, Nazih Ziade, Caspar Horne, XiChuen Du, Hans du Buf, Mohammad Kardan, Daniel Cano, Thien Haminh, Gou Youedong, Jie Wang et notre chère secrétaire Heidi Bernau.

Un dernier merci à tous mes amis et membres de ma famille qui ont dû me supporter pendant ces derniers mois.

Résumé

Ce travail de doctorat traite le problème de la segmentation des images en un ensemble de régions correspondant autant que possible aux objets de la scène.

L'objectif de cette segmentation est de passer d'une représentation numérique à une représentation symbolique des images, c'est-à-dire les régions et les paramètres qui les caractérisent. On passe ainsi d'un modèle local constitué par l'ensemble des échantillons numériques de l'image à un modèle plus élaboré où les frontières des régions constituent les contours des objets de l'image et le signal luminance à l'intérieur de chaque région leur texture.

Une telle représentation peut trouver des applications en interprétation de scène et en codage d'images, ceci à cause de la validité du modèle contour-texture utilisé en relation avec les propriétés du système visuel humain. Nous avons étudié le potentiel de la méthode de segmentation dans le contexte de la réduction de redondance appliquée aux images.

La méthode de segmentation proposée procède de manière adaptative. Etant donné une approximation pour modéliser la texture à l'intérieur de chaque région, on cherche à adapter d'une façon optimale les paramètres de cette approximation à la forme des régions. Pour y parvenir, l'image est décomposée dans un premier temps en un ensemble de carrés de différentes tailles de manière à obtenir une correspondance aussi grande que possible entre le signal original et son approximation à l'intérieur de chaque carré. Dans un deuxième temps, ces carrés sont itérativement rassemblés jusqu'à atteindre un certain nombre de régions de forme quelconque. On regroupe successivement les deux régions les plus ressemblantes selon la correspondance qui existe entre le signal original et son approximation sur la région obtenue après rassemblement.

On a étudié les résultats de cette méthode en représentant la luminance de chaque région par des fonctions polynômiales de degré 0 à 3. Une fois la segmentation finale obtenue, les informations de frontières de régions et d'approximation sont codées séparément.

Les performances en compression d'information sont remarquables: des images de qualité tout à fait acceptable peuvent être obtenues avec des taux de compression dépassant 30 à 1. Le contenu sémantique essentiel d'une scène peut être préservé pour des facteurs de compression allant de 60 à 1 à 130 à 1 selon le genre d'image traitée.

Abstract

ADAPTIVE SEGMENTATION FOR IMAGE CODING.

This Ph.D. dissertation analyses the problem of segmenting an image into a set of regions corresponding as much as possible to the real objects of a scene.

The goal of this segmentation is to go from a numerical representation of an image to a symbolic one, i.e. the regions and their characteristic features. Starting from the set of picture elements, one reaches a more compact model where region frontiers define the contours of the objects and the signal within each region their texture.

Such a representation can find useful applications for scene understanding and image coding. Recent works have shown the potential of a contour-texture model for image coding due to the properties of our human visual system. We have studied here the way of coding a segmented representation of an image when using high order polynomials for approximating the different regions.

The proposed segmentation algorithm is adaptive. Given a certain approximation to model the texture within each region, one tries to modify adaptively the region shape and the approximation parameters. To do so, the image is first split into a set of squares of different sizes in order to obtain an optimal correspondance between the original signal and its approximation within each square. Then starting from this initial partition, adjacent regions are iteratively merged till one reaches a segmentation with a certain number of regions of any shape. At each step of the merging process, the two most similar regions are associated on the basis of the adequacy of the approximation on the new region.

Polynomials of degree 0 to 3 have been used in the approximation process. Once the final segmentation is obtained, frontier information and texture information are coded separately.

Performances for redundancy reduction are impressive: acceptable quality images can be obtained with compression ratios of the order of 30 to 1. It is shown how most of the semantics can be preserved with coded pictures at compression ratios ranging from 60 to 1 to 130 to 1. The algorithm has been applied to three different 256x256 natural images quantized with eight bit dynamics.

Table des matières

<u>Remerciements</u>	iii
<u>Résumé</u>	iv
<u>Abstract</u>	v
<u>Tables des matières</u>	vi-x
1 Introduction	1-7
<i>1.1 Segmentation et compression d'images</i>	1-5
1.1.1 Codage d'images à haute compression (méthodes de deuxième génération)	2-3
1.1.2 Historique des méthodes de segmentation	3-5
- Formation de frontières	4
- Formation de régions	4-5
<i>1.2 Contribution de notre travail</i>	5-6
<i>1.3 Organisation de la thèse</i>	6-7
2 Segmentation adaptative	8-45
<i>2.1 Modèle contour-texture</i>	8-21
2.1.1 Buts visés	11-15
- Commentaires	14-15
2.1.2 Approche adaptative	15-16
2.1.3 Adaptation des paramètres du modèle	16
2.1.4 Adaptation de la forme des régions: Division et Rassemblement	16-20
2.1.5 Complexité du problème	20-21
<i>2.2 Extraction d'images de contrôle</i>	22-33
2.2.1 Hypothèses sur les images traitées	22
2.2.2 Extraction de contours	22-32
2.2.3 Images des structures positives et négatives	32-33
<i>2.3 Approximation des données</i>	34-43
2.3.1 Type d'approximation	34-40
- Considérations générales	35-37
- Critère d'approximation minimax	37-38
- Effet du paramètre p	38-39
- Effet des coefficients de pondération $a(k,l)$	39
- Approximation au sens des moindres carrés	39-40
2.3.2 Fonctions d'approximation	40-41

2.3.3 Quantification de l'approximation	42-43
<i>2.4 Mesures de qualité</i>	43-45
2.4.1 Mesures globales	43-44
2.4.2 Mesures contextuelles	44-45
- Commentaires	45
<i>2.5 Conclusion</i>	45
3 Opération de division	46-93
<i>3.1 Principes généraux</i>	46-49
3.1.1 Existence d'une solution dans le calcul de l'approximation	48
3.1.2 Formes des régions après division	48-49
<i>3.2 Implémentation de l'algorithme</i>	49-51
3.2.1 Analyse hiérarchique	50
3.2.2 Analyse descendante-ascendante cyclique	51
<i>3.3 Complexité de l'algorithme de division</i>	51-59
3.3.1 Aspects liés à l'approximation	52-57
- Rôle de la séparabilité	55-57
3.3.2 Simplifications rendues possibles grâce à la structure des images traitées	57-59
<i>3.4 Représentation de l'information segmentée et postraitement</i>	60-67
3.4.1 Position et taille des carrés	60-65
- Codage par arbre quaternaire	60-61
- Codage conditionnel par taille de carré	61-65
3.4.2 Quantification des coefficients polynômiaux	65-66
3.4.3 Postraitement	66-67
<i>3.5 Analyse expérimentale de l'algorithme de division</i>	67-92
- Convention	68
3.5.1 Division effectuée sans image de contrôle des contours	68-72
3.5.2 Effets des paramètres et compression	73-80
- Rôle du seuil sur l'erreur quadratique moyenne	80-84
- Rôle du degré du polynôme	84-86
- Aspects liés à la compression d'information	86-90
3.5.3 Limitation de la plus petite taille de carré	90-92
<i>3.6 Conclusion</i>	92-93
4 Opération de rassemblement	94-139
<i>4.1 Principe de l'algorithme</i>	94-101
4.1.1 Précision dans la description des frontières de région	95-97
4.1.2 Choix des critères de dissimilarité entre régions	97-100
- Mesure de distance entre coefficients d'approximation u_i sur chaque région	97

- Erreur quadratique (moyenne) par rapport au signal original	97-98
- Erreur quadratique (moyenne) entre signal approximé avant regroupement et signal approximé après regroupement	98-100
- Erreur quadratique courante pondérée par le nombre de points de chaque région	100
- Remarque	100
4.1.3 Critères d'arrêt du processus	101
Commentaire	101
<i>4.2 Evolution de l'approximation après rassemblement</i>	101-106
4.2.1 Décomposition de l'approximation au sens des moindres carrés	101-102
- Commentaires	103-104
4.2.2 Application à l'algorithme de rassemblement	104-105
4.2.3 Evaluation des critères de dissimilarité	105-106
<i>4.3 Structure des données et implantation du GCR</i>	107-112
4.3.1 Réalisation des noeuds du GCR	107-108
4.3.2 Réalisation des branches du GCR	108-109
4.3.3 Mise à jour et ordonnancement des mesures de dissimilarité	109-110
4.3.4 Sructure BRAIN	110-112
<i>4.4 Complexité de calcul et solutions sous-optimales</i>	113-117
4.4.1 Complexité de l'algorithme	113-115
4.4.2 Simplification du calcul d'approximation	115-116
4.4.3 Solutions sous-optimales par modification du processus de rassemblement	116-117
<i>4.5 Stratégie globale de rassemblement</i>	117-138
4.5.1 Analyse des défauts	117-130
4.5.1.1 Analyse expérimentale des effets des critères de dissimilarité	118-127
4.5.1.2 Distorsions causées par l'augmentation du degré du polynôme	128-130
4.5.2 Regroupement des régions sans contours	130-132
4.5.3 Insertion dans le GCR des régions adjacentes aux contours	132-135
4.5.4 Rassemblement final	135-138
<i>4.6 Conclusion</i>	139
5 Représentation de l'information segmentée et postraitement	140-167
<i>5.1 Codage de l'information contour</i>	140-154
5.1.1 Codage de frontière par image interfrontière	143-145
5.1.2 Elimination des liens de régions	145-149
5.1.3 Algorithme de codage du graphe de contours	149-154
<i>5.2 Codage de l'information texture</i>	155-164
5.2.1 Coût maximal des coefficients d'approximation	155-157
5.2.2 Quantification vectorielle des coefficients polynômiaux	157-162

- Formulation du problème	157-158
- Algorithmes de coalescence	158-159
- Mesures de distorsions	159-162
5.2.3 Variation du nombre de coefficients par région	162-163
5.2.4 Stratégie globale	163-164
<i>5.3 Elimination des faux contours</i>	165-167
<i>5.4 Conclusion</i>	167
6 Résultats en codage d'images	168-231
<i>6.1 Paramètres d'expérimentation et calcul de compression</i>	168-175
6.1.1 Degré du polynôme	168-169
6.1.2 Nombre de régions	169-170
6.1.3 Autres paramètres intervenant dans la qualité des résultats	170-173
6.1.4 Facteur de compression	173-175
<i>6.2 Résultats expérimentaux et discussion</i>	175-214
6.2.1 Segmentation adaptative par polynômes de degré 0	175-185
6.2.2 Segmentation adaptative par polynômes de degré 1	185-193
6.2.3 Segmentation adaptative par polynômes de degré 2	194-203
6.2.4 Segmentation adaptative par polynômes de degré 3	204-214
<i>6.3 Effet du prétraitement de l'image originale</i>	215-218
<i>6.4 Améliorations</i>	219-231
6.4.1 Segmentation adaptative avec modification du degré de polynôme	219-226
6.4.2 Estimation de la composante aléatoire	227-231
7 Conclusion générale	232-234
<i>7.1 Principe de la segmentation adaptative</i>	232
<i>7.2 Résultats obtenus dans le contexte du codage d'images</i>	232-233
<i>7.3 Problèmes ouverts et améliorations</i>	233-234
Annexe A: Minimisation de la fonction erreur d'approximation $A(\mathbf{e}) = \mathbf{e}^T A \mathbf{e}$	235
Annexe B: Modification de l'approximation au sens des moindres carrés par orthogonalisation de l'espace des fonctions de base	236
Annexe C: Existence d'une solution polynômiale optimale au sens des moindres carrés sur des domaines carrés de dimension paire	237-238
Annexe D: Nombre d'opérations nécessaires au calcul des produits matriciels de l'équation (3.12)	239-240

<u>Annexe E:</u> Estimation de l'erreur quadratique lors d'une formulation séparable du problème d'approximation - Application aux polynômes séparables	241-242
<u>Annexe F:</u> Modification de la liste des voisins après rassemblement	243-244
<u>Annexe G:</u> Expression des paramètres d'approximation par changement d'origine	245-246
<u>Annexe H:</u> Invariance de la distance de Mahanalobis par changement d'origine pour des vecteurs de coefficients polynômiaux	247
<u>Références</u>	248-252
<u>Curriculum Vitae</u>	253

1. Introduction

1.1 Segmentation et compression d'images :

Avec les sons, les images constituent pour l'être humain les moyens les plus importants de percevoir la réalité du monde environnant. La perception de la lumière, puis le sens de la vision se développent dès sa naissance, pour aboutir à un système visuel complexe. Ce système est l'étape d'une évolution de plusieurs millions d'années. Ses performances sont impressionnantes.

Notre rétine reçoit de multiples informations chaque seconde. Ces informations sont combinées, intégrées, comparées tout au long d'une chaîne de traitement qui permet de voir puis d'interpréter la scène qui s'offre à nos yeux. La distinction de formes simples tels lignes droites, angles droits, cercles... s'effectue au niveau des cellules simples, complexes et hypercomplexes du cortex visuel. Cependant, l'architecture et le fonctionnement de notre système nerveux nous sont encore trop inconnus pour qu'on puisse comprendre comment s'effectue la phase d'interprétation en elle-même.

Aussi, il est utopique de vouloir réaliser un système de vision par ordinateur qui puisse avoir des performances comparables à celles de notre système visuel. Si l'homme ne peut encore connaître le fonctionnement neurophysiologique de son cerveau, la nature lui a donné une capacité créatrice à partir de ce qu'il perçoit. Il sait analyser, comprendre, modéliser, inventer... D'après les grands philosophes grecs tels Platon, il existe au niveau absolu un concept, une idée. Pour comprendre le fonctionnement du monde, l'homme bâtit des modèles qui vont se rapprocher tant que possible de cette idée. Le modèle peut se situer au niveau de la représentation de la réalité, en l'occurrence l'image. Il peut se faire en termes d'objets connus. Un modèle de la connaissance est nécessaire pour caractériser les objets détectés dans une image. Ces opérations se situent au niveau de ce qui est appelé le traitement d'images de haut niveau. Dans un premier temps, il est cependant nécessaire de décomposer l'image en un ensemble de régions qui correspondent plus ou moins à des objets. Cette opération que l'on appellera la segmentation fait partie du traitement d'images de bas niveau. Elle constitue l'une des étapes fondamentales de l'analyse de scène.

Une segmentation idéale associe à chaque objet de la scène une région. En réalité, celle-ci ne peut être obtenue qu'une fois que l'on a compris le contenu de la scène. Dans un traitement de bas niveau, il s'agit plus d'associer des parties d'objets correspondant à des surfaces homogènes pour constituer une première segmentation. C'est l'un des objectifs de ce travail. Plus précisément, on va chercher à créer une représentation d'une image en un ensemble de régions dont on peut modéliser de façon plus ou moins approchée la luminance par un certain nombre de paramètres. La partition de l'image se fera par une adaptation conjuguée de la valeur de ces paramètres et de la forme de chaque région.

A partir d'une telle représentation, on peut essayer d'interpréter le contenu de l'image. Il apparaît cependant ambitieux de vouloir combiner dans un seul travail le traitement d'image de haut et de bas niveau. Cependant, les symboles que constituent les objets ou les parties d'objet d'une image obtenus par segmentation constituent une représentation très compacte de l'information. Pourquoi ne pas essayer de trouver des applications de la segmentation à la compression

d'information dans les images? A la suite de travaux récents en codage d'images [1], il est apparu qu'on pouvait réduire l'information à transmettre de plusieurs ordres de grandeurs en adoptant une représentation contour-texture d'une image [2], tout en conservant l'essentiel du contenu informationnel de l'image. En définissant une méthode de segmentation qui s'inscrit parfaitement dans le cadre de ce modèle, ses performances en codage d'images se doivent d'être analysées vu l'impact toujours considérable de ce sujet.

1.1.1 Codage d'images à haute compression (méthodes de deuxième génération):

Le but du codage d'images est de minimiser autant que possible la quantité d'information (nombre de bits) nécessaire à une représentation fidèle d'une image originale. Quel que soit le système d'acquisition d'une image, la représentation numérique qui en résulte (appelée forme canonique de l'image) correspond à une suite d'échantillons de luminance régulièrement espacés et quantifiés en un certain nombre de valeurs appelées niveaux de gris. Le pas d'échantillonnage est en général choisi suffisamment petit pour éviter d'interpoler les échantillons lors la reconstitution du signal analogique, c'est-à-dire l'affichage. Il utilise ainsi à bon escient la capacité d'intégration de l'oeil humain. Une image numérique constitue ainsi un tableau bidimensionnel de nombres.

Le contenu de ce tableau n'est en général pas aléatoire. En effet, des échantillons voisins ont des valeurs comparables, ce qui correspond à une importante corrélation spatiale. En exploitant cette corrélation, on peut représenter d'une façon compacte ces valeurs. La plupart des techniques de codage d'images dites de première génération [3] ont ainsi essayé de tirer profit de cette corrélation pour maximiser la compression d'information.

En se limitant à cette approche, on ne s'est pas placé pour autant au niveau de l'entropie de l'image qui est souvent inconnue et dépend fortement du modèle utilisé. En effet, les techniques classiques de codage admettent comme implicite le fait que l'on ait à faire à des signaux stationnaires et ergodiques, chose qui est rarement satisfaite dans le contexte d'images réelles. Si les méthodes de première génération ont mis l'accent sur la façon de coder les messages extraits d'une image, aucun effort n'avait été porté alors quant au choix de ces messages. Les méthodes de deuxième génération ont mis l'accent sur ce point.

Les images traitées ou codées par ordinateur sont en général destinées à un observateur humain. Même si ce n'est le cas, ne faut-il pas essayer de réaliser grâce à l'informatique des analyses ou des traitements comparables à ceux qu'effectue notre système visuel? Au lieu d'associer les échantillons d'une image numérique arbitrairement pour prédire la valeur de l'échantillon suivant ou pour calculer un ensemble d'attributs (coefficients transformées) sur cette série d'échantillons, ne peut-on pas dissocier contours et textures d'une image, comme l'ont proposé Kocher et Kunt [2]? Les contours correspondent aux endroits de variation brusque de luminance. Selon la rapidité du saut de luminance et son amplitude, il est possible de séparer les contours en contours faibles et contours forts. Quant aux textures, la notion correspondante est liée à la perception visuelle et elle relève de beaucoup d'aspects subjectifs [4]. Il n'existe pas de définition communément admise les caractérisant. On peut admettre qu'une texture est invariante par translation [5]. Dans le contexte du codage d'images, les textures représentent tout ce qui n'est pas contour. Réaliser une compression élevée peut

être obtenu à partir d'une décomposition d'une image en termes de régions à texture uniforme entourées par des contours. Si les contours sont fermés, on aboutit à un modèle segmenté de l'image tel celui de Kocher et Kunt [2], modèle dans lequel s'inscrit notre travail et qui sera élaboré au chapitre 2.

Si les frontières des régions ne sont pas fermées, on peut décomposer l'image en une composante basse-fréquence caractérisant l'information de texture et une composante haute-fréquence. Cette dernière peut être codée telle quelle en utilisant la réponse impulsionnelle du filtre haute fréquence; il s'agit de la méthode des "Intensités Artificielles" [6]. Une autre possibilité consiste à séparer la partie haute fréquence en une série de filtres directionnels [7,8]. Ceux-ci vont permettre de caractériser les contours de l'image suivant leur orientation. Les contours dans chaque direction peuvent être sous-échantillonnés vu qu'ils ont été détectés par des filtres directionnels limités en bande. Le profil du signal haute fréquence de part et d'autre de chaque contour est modélisé par des fonctions paramétriques unidimensionnelles à 2 ou 3 variables (polynômes, gaussienne) ou par l'amplitude du profil et la réponse impulsionnelle du filtre somme des filtres directionnels, c'est-à-dire le filtre haute-fréquence (Intensités Artificielles).

Carlsson a proposé récemment une technique où il est possible de reconstruire une image en se basant uniquement sur l'information de contour [9]. Dans un premier temps, les contours importants sont détectés en utilisant un masque Laplacien adouci. L'information de niveau de gris de part et d'autre de ces contours ainsi que leur position sert au codage. Dans toutes les zones ne correspondant pas à des contours, on reconstitue le signal en résolvant une équation de Laplace par une méthode itérative de relaxation. Les conditions aux limites sont imposées par la valeur des niveaux de gris situés de part et d'autre des contours. La résolution d'une équation de Laplace est équivalente à imposer un minimum de variation de luminance dans les zones de texture. Le résidu, c'est-à-dire la différence entre l'image originale et l'image reconstruite est codé par codage pyramidal [10].

Après avoir placé le contexte des techniques de codage de deuxième génération basées sur un modèle contour-texture des images, faisons un historique des méthodes de segmentation.

1.1.2 Historique des méthodes de segmentation:

Un large nombre de techniques de segmentation ont été proposées tout au long du développement des méthodes d'analyse de scène. La plupart des approches se sont limitées à des applications particulières (par exemple, [11,12]) ou se sont fixées des buts bien définis (par exemple, [13]). Il n'existe encore aucun système capable de caractériser les objets de n'importe quelle image naturelle d'une façon parfaite.

On distingue les méthodes de segmentation dites de formation de frontières des méthodes de formation de régions. Les algorithmes permettant d'extraire les frontières impliquent des opérations de différentiation, de recherche de maxima et de regroupement de segments locaux en contours. Les approches visant à former des régions comprennent la croissance de région par analyse locale, l'analyse de propriétés globales.

Formation de frontières:

En appliquant des opérateurs de différentiation spatiale à une image, on peut mesurer l'importance de son gradient. En plaçant un seuil sur ce dernier, on obtient une image formée d'un ensemble de contours séparés. Si cela s'avère nécessaire, il faut éliminer les contours peu importants ou les contours fictifs (c'est-à-dire ceux qui correspondent à des réponses multiples de l'opérateur de différentiation utilisé). Les contours restants sont alors associés pour représenter les frontières de régions.

Une grande partie des débuts de la recherche en analyse de scène était basée sur la recherche d'algorithmes de poursuite de lignes droites [14,15]. En se limitant à des images composées d'objets pouvant être assimilés à des polyèdres, plusieurs auteurs ont proposé d'utiliser la position des sommets de ces polyèdres pour relier les différents segments détectés [16-18]. La contrainte imposée de n'avoir que des formes polygonales rendait l'utilisation seule de ces méthodes inefficace pour la segmentation d'images naturelles. Mackworth [19] puis Horn [20] ont essayé de baser la définition de leurs algorithmes de poursuite de frontières sur une analyse des surfaces de luminance (leur orientation, leur courbure, la lumière qu'elles réfléchissent). Il est possible de poursuivre un contour en recherchant les points de valeur comparable de gradient. Certains auteurs ont proposés d'extraire globalement certaines informations relatives aux contours pour pouvoir faciliter leur association [21,22]. Ainsi, des contours possédant des propriétés directionnelles similaires (estimées globalement) peuvent être utilisées au moment d'une analyse locale pour effectuer la formation de frontières.

Zucker, Hummel et Rosenfeld ont suggéré [23] d'utiliser des procédures de relaxation pour la formation de frontières. L'avantage de cette méthode consiste à considérer toutes les orientations possibles en différents points contours voisins d'un certain point contour pour définir l'orientation la plus probable du contour en ce point.

Quelle que soit la démarche choisie, aucune méthode de formation de frontières ne peut donner naissance à des frontières ayant toutes une signification sémantique au niveau de l'interprétation. Il en va de même au niveau des régions obtenues par les techniques de formation de régions que l'on va décrire ci-dessous.

Formation de régions:

Il est possible de décomposer n'importe quelle image en un ensemble de régions en regroupant tous les points images qui possèdent une certaine propriété avec une tolérance n'excédant pas un certain seuil. On utilise en général un balayage successif de tous les points voisins d'une certaine région pour les inclure à cette région. Cette approche générale est connue sous le nom de croissance de région. Le seuil de tolérance ne peut être fixé une fois pour toute pour une certaine image si l'on désire obtenir des régions se rapprochant des objets de la scène. Toutes sortes de solutions ont été proposées pour fixer le seuil de tolérance de façon dynamique mais il est difficile de le faire si l'on n'estime pas des paramètres plus globaux et si l'on n'a pas de connaissance a priori de la scène que l'on désire segmenter. Freuder [24] a introduit une méthode relativement intéressante pour

regrouper des régions en effectuant une mesure de similarité entre régions voisines et en décidant de regrouper les deux qui sont les plus ressemblantes. Plus tard, Horowitz et Pavlidis [25] ont suggéré de limiter l'analyse de toutes les configurations possibles de régions à comparer deux à deux en introduisant le concept de division et rassemblement. L'image est subdivisée dans un premier temps en zones homogènes de forme simple (par exemple, des carrés de différentes tailles). Ces carrés sont alors comparés pour être rassemblés jusqu'à obtenir des régions de forme quelconque.

Une approche plus récente encore consiste à effectuer la formation de régions sur la base d'une analyse sémantique. A cet égard Levine et Nazif [26] ont construit un système expert. A notre avis, les résultats obtenus restent cependant comparables aux méthodes de croissance de région [27] ou de division et rassemblement [25] classiques.

Certains auteurs ont enfin suggéré de former des régions en effectuant des mesures de caractéristiques d'une image. Olhander [28] montre ainsi comment découper itérativement une image en plaçant des seuils aux minima des histogrammes de différents attributs extraits de l'image. Malheureusement, ce genre de méthode ne tient pas bien compte des relations spatiales des différentes régions. A cet effet, Hanson, Riseman et Nagin [29] ont décrits comment les prendre en considération en estimant les différents attributs sur des fenêtres de différentes tailles.

Analysons maintenant en quoi notre travail se distingue des méthodes présentées jusqu'ici.

1.2 Contribution de notre travail :

Notre travail aborde en premier lieu le problème de la segmentation d'une scène naturelle. La méthode de segmentation proposée s'intègre dans le contexte des méthodes de formation de régions. Elle introduit cependant le concept de *segmentation adaptative*. Dans un premier temps, on définit un modèle souple basé sur des approximations polynomiales pour représenter différentes configurations de luminance. Dans un deuxième temps, on effectue un arrangement optimal entre les paramètres du modèle polynomial servant à décrire le signal luminance à l'intérieur de chaque région et la forme de ces régions. C'est à ce niveau qu'apparaît l'adaptation. L'algorithme de formation de régions se base sur la philosophie de division et rassemblement introduite par Horowitz et Pavlidis [25].

En montrant l'importance de l'information de contour dans les images, des contraintes sont imposées à la méthode de segmentation de manière à préserver ce genre d'information. Dans la segmentation finale, les frontières des régions vont épouser autant que possible les contours de l'image. De ce fait, les régions obtenues se rapprochent d'objets ou de parties d'objets constituant la scène. L'essentiel du contenu sémantique de la scène est ainsi préservé. Il est donc tout à fait envisageable de comparer les régions obtenues à des objets connus pour effectuer une interprétation de l'image. Celle-ci ne sera pas abordée ici mais un mode de représentation des frontières de régions et du modèle d'approximation de luminance va permettre d'utiliser les résultats de la segmentation dans le contexte du codage d'images.

A cause du grand nombre de paramètres que l'on peut utiliser pour aboutir à la segmentation finale, la méthode de codage proposée s'avère efficace pour représenter avec plus ou moins de distorsion une scène originale. En limitant le nombre de régions de la segmentation finale, on augmente la compression mais on élimine les objets de petite taille de l'image. Des résultats de bonne qualité sont obtenus pour des images originales 256x256 à 8 bits comprimées d'environ un facteur de compression de 30 à 1. Dans certains cas, pour des images constituées de grandes plages uniformes, l'information peut être réduite d'un facteur 130 à 1 sans que le contenu essentiel de l'image ne soit perdu et tout en garantissant une précision absolue au niveau des contours réels des objets apparaissant dans l'image segmentée.

Ce travail définit donc une méthode de segmentation adaptative qui garantit une bonne correspondance entre les objets réels d'une scène et les régions formées. La sophistication du modèle contour-texture proposé permet d'utiliser les résultats de la segmentation dans le contexte du codage d'images de moyenne à haute compression selon le degré de détail désiré. On pourra comparer les résultats obtenus aux autres méthodes de codage d'images de deuxième génération (voir paragraphe 1.1.1).

1.3 Organisation de la thèse :

Le chapitre 2 présente la philosophie de la méthode de segmentation adaptative. Il introduit le modèle contour-texture et définit les objectifs visés par l'algorithme de segmentation dans le contexte du codage d'images. Il présente comment adapter de façon conjuguée les paramètres du modèle de représentation de chaque région à la forme de ces régions. Le tout aboutit à un algorithme adaptatif de division et rassemblement dont les détails sont mis en évidence aux chapitres 3 et 4. Ce chapitre décrit également comment extraire des images de contrôle liées aux contours servant à piloter le processus de segmentation. Il justifie le choix du critère des moindres carrés pour estimer les paramètres d'approximation polynomiale du signal à l'intérieur de chaque région. Enfin, il propose des mesures servant à faire évoluer la segmentation jusqu'à obtenir le résultat désiré.

Le chapitre 3 aborde le détail de la première phase de l'algorithme de segmentation: la division adaptative de l'image en des carrés de différentes tailles sur la base d'une correspondance optimale entre le signal d'approximation à l'intérieur de chaque carré et l'original. L'accent est mis sur la non dégradation de l'information contour. Ce chapitre présente également des détails d'implantation quant à la structure des données et à la complexité algorithmique de l'opération de division. Il montre comment représenter sous forme d'un arbre quaternaire l'information à ce niveau de la segmentation. Des résultats préliminaires de codage donnent une idée de la réduction de redondance et de la qualité des images qu'on est en mesure de reconstituer après division.

Le chapitre 4 décrit la deuxième phase de l'algorithme de segmentation: le rassemblement adaptatif des régions obtenues après division. En construisant un graphe permettant de comparer toutes les associations possibles de deux régions adjacentes, on choisit de regrouper d'une façon itérative les deux régions qui introduisent le moins de distorsion au niveau de l'approximation après

regroupement. Pour pouvoir estimer la distorsion introduite, différentes mesures d'erreur sont proposées. Du fait d'une propriété de décomposition de l'approximation au sens des moindres carrés, on montre comment calculer ces mesures rapidement de manière à rendre possible l'implantation de l'algorithme de rassemblement. Comme au chapitre 3, des problèmes de structure de données et de complexité algorithmique sont abordés dans ce chapitre. Enfin, ce chapitre définit à partir d'une procédure expérimentale comment combiner les mesures de distorsions pour s'approcher au mieux d'une segmentation idéale constituée uniquement d'objets de la scène.

Dans le chapitre 5, on décrit la manière de coder chaque type d'information à partir d'une image segmentée, c'est-à-dire information contour et information texture. Les frontières des régions vont être décrites sans erreur au moyen d'un code de chaîne après avoir légèrement modifié la segmentation de manière à imposer des contraintes à la succession des symboles pouvant exister dans le code. L'information texture est caractérisée par des surfaces de reconstruction polynômiales. On montre comment coder les coefficients de ces polynômes. La même surface de reconstruction peut se rapporter à des parties non connexe de l'image grâce à l'utilisation de techniques de quantification vectorielle. Enfin, ce chapitre illustre comment améliorer la qualité de l'image reconstruite en éliminant les faux contours.

Le chapitre 6 donne les résultats de la méthode dans le contexte du codage d'images. Des images codées avec des facteurs de compression allant de 30 à 1 jusqu'à 130 à 1 y sont présentées. On analyse l'effet du nombre de paramètres d'approximation (c'est-à-dire le nombre de coefficients polynômiaux) sur la qualité de la segmentation finale. On montre comment on peut encore améliorer cette qualité en combinant des approximations par des constantes avec des approximations de degré plus élevé. Ce chapitre discute également de l'effet du prétraitement de l'image originale par des opérateurs non linéaires d'élimination de granularité tel que le filtre de Nagao.

Le chapitre 7 sert de conclusion à ce travail. Il montre les problèmes qui restent ouverts en analyse et interprétation de scène et suggère des améliorations qui pourraient augmenter la qualité des résultats obtenus.

Le lecteur désireux d'avoir un aperçu des résultats que l'on peut obtenir peut se limiter aux chapitres 2, 5 et 6. Le chapitre 2 donne les bases de la méthode de segmentation. Le chapitre 5 indique comment coder l'information segmentée alors que le chapitre 6 décrit et discute les résultats de codage. Les chapitres 3 et 4 mettent en évidence l'implantation de l'algorithme de segmentation proposé. Ils décrivent le détail de propriétés mathématiques nécessaires à la réalisation de la méthode ainsi que comment combiner tous les paramètres pour aboutir au résultat désiré.

2. Segmentation adaptative

Ce chapitre présente une méthode assez générale et adaptative de segmentation d'images de manière à faire ressortir les structures de la scène. Il s'agit essentiellement de définir les critères permettant de caractériser une image et donc de la modéliser. L'accent sera mis sur la nécessité d'une analyse contextuelle tenant compte des propriétés statistiques et structurelles de l'image. Trois idées essentielles sont à retenir :

- l'image est représentée par un *ensemble de régions*. A chacune de ces régions est associée une certaine approximation permettant de décrire la variation de son niveau de gris.

- la forme de ces régions est définie par l'*adaptation optimale* des paramètres d'approximation à la structure des données.

- la nécessité d'une *approche évolutive* pour déterminer la taille, le nombre et la forme des régions, vu que l'analyse de l'image se fait à toute une série de niveaux pour lesquels les règles sémantiques sont différentes.

Ce chapitre se divise en cinq sections : La première établit un modèle des images et en justifie le choix d'une représentation segmentée. La deuxième traite de l'extraction d'images de contrôle à partir de la scène originale, images qui permettront de piloter le processus de segmentation. La section suivante analyse le problème de la représentation des données pour chaque région et insiste sur la nécessité d'effectuer la partition de l'image en tenant compte du choix qui a été fait pour approximer les données. La quatrième section définit les mesures de qualité qui seront utilisées pour déterminer l'adéquation entre la représentation choisie et le signal de départ; l'accent sera porté sur la nécessité d'une analyse à la fois globale et contextuelle. Enfin, le chapitre se termine par un rappel des points essentiels présentés.

2.1 Modèle Contour-Texture :

Une fonction réelle positive de deux variables x et y , notée $g(x,y)$, définit une représentation mathématique d'une image à niveaux de gris. Sa valeur au point (x,y) , dite niveau de gris, est proportionnelle à l'intensité lumineuse en ce point. En fait, n'importe quel moyen permettant de caractériser l'évolution de cette fonction réelle sur l'étendue finie de l'image constitue un modèle de celle-ci. Nous appellerons "image naturelle" une scène habituelle du monde qui nous entoure, telle un paysage de montagne, un portrait, une ville... Néanmoins, il est souvent difficile, voire impossible de trouver une expression analytique permettant de décrire l'évolution de n'importe quelle image naturelle. Une représentation plus courante et relativement simple consiste à mettre l'image sous sa forme numérique. Il suffit pour cela d'échantillonner la fonction $g(x,y)$ en un certain nombre de points et de quantifier la valeur échantillonnée selon un nombre fini d'amplitudes entre le niveau 0 (noir) et le niveau maximum (blanc). Pour éviter toute perte d'information, les images doivent être échantillonnées en respectant le théorème de Shannon, c'est-à-dire que la fréquence d'échantillonnage soit au moins deux fois supérieure à la fréquence maximale du signal image dans chaque direction. Bien entendu, on peut être moins restrictif en supposant que les effets de recouvrement du spectre

du signal image ne tombent pas dans la bande passante de l'oeil humain, pour autant que l'image soit destinée à un tel observateur. Par la suite, on ne considèrera que des images mises sous forme numérique $g(k,l)$. La grille d'échantillonnage sera supposée rectangulaire et constituée de M lignes \times N colonnes. Le processus de numérisation est présenté à la figure 2.1 (les signaux sont présentés à une dimension pour simplicité).

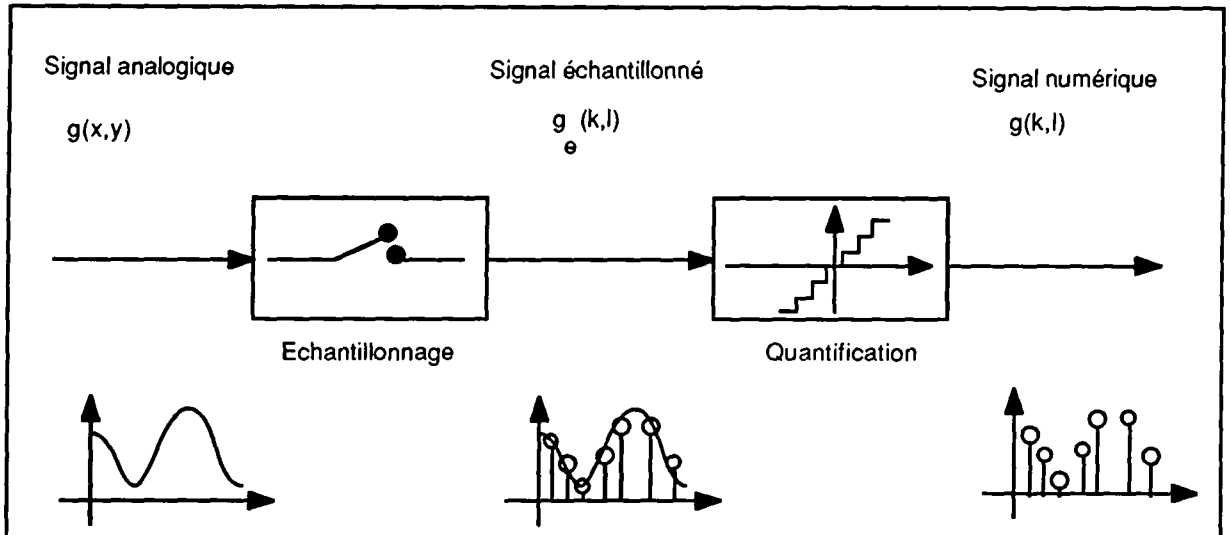


Figure 2.1 : Processus de numérisation d'une image
(les signaux sont présentés à une dimension pour des raisons de simplicité)

Les figures 2.2, 2.3 et 2.4 présentent les trois images naturelles 256×256 à 8 bits qui seront utilisées dans le cadre de ce travail. L'image 2.2 (bâtiment) a été choisie parce qu'elle présentait des caractéristiques géométriques intéressantes. L'image 2.3 (caméraman) possède une partie texturée (herbe) et une partie dont l'évolution du niveau de gris est relativement complexe (caméra). De plus, cette image est très contrastée et présente des détails aussi bien dans les parties claires que dans les parties sombres. L'image 2.4 (couple) est intéressante car elle représente des visages humains. Vu l'habitude que nous avons à côtoyer des êtres humains chaque jour, il est raisonnable de penser que nous sommes particulièrement sensibles aux distorsions apportées à ce genre d'images. Ces images ont été choisies également parce qu'elles permettront de mieux apprécier l'évolution des résultats de codage par rapport au travail de Kocher [27].



Figure 2.2 : Image du bâtiment.

Figure 2.3 : Image du caméraman.



Figure 2.4: Image du couple

La représentation numérique $g(k,l)$ peut déjà être considérée comme un modèle dont les paramètres sont les points images ou pixels. Ce modèle a l'avantage d'être très général mais possède le grave inconvénient de faire intervenir un nombre énorme de paramètres (les pixels), chacun très local.

La perception humaine, en revanche, utilise d'autres paramètres. Par exemple, un dessinateur commence son dessin en esquissant les pourtours des objets et remplit ensuite les régions ainsi délimitées. Les travaux récents de neurophysiologie [30] montrent clairement l'importance du rôle joué par les contours dans la perception. Il est donc légitime d'utiliser un modèle général de contour-texture. Ici, la texture fait référence à tout ce qui n'est pas contour. On peut

y ajouter éventuellement une troisième composante, la structure, pour lier les régions les unes aux autres.

Puisque les contours sont des paramètres de ce modèle, il faut les extraire d'une façon ou d'une autre à partir des données $g(k,l)$. Il existe deux approches générales à l'extraction de contour : celle qui vise à les obtenir directement et celle qui cherche à déterminer les régions de texture uniforme dont les frontières définissent les contours.

Kocher, en adoptant la deuxième méthode [27] a montré son potentiel énorme pour la compression d'information. Toutefois, la généralisation directe de cette approche pour en augmenter davantage les performances, conduit à une complexité très déraisonnable. C'est pourquoi, notre approche se démarque de celle de Kocher et cherche à atteindre les buts suivants.

2.1.1 Buts visés :

La vision par ordinateur simule le fonctionnement de notre système visuel. Il s'agit de trouver les objets d'une image et de caractériser ceux-ci selon une certaine approximation. Cela ne correspond-il pas au processus quotidien de chacun d'entre nous lorsqu'il regarde une scène naturelle? Comme indiqué par Riseman et Arbib [31], la complexité impliquée par sa conception et sa réalisation a conduit à décomposer le problème en toute une série de démarches dites de "bas niveau" et de "haut niveau" (voir chapitre 1). La partition de l'image en une série de régions, c'est-à-dire la segmentation, constitue une étape préliminaire à l'opération d'interprétation. En effet, celle-ci nécessite le plus souvent la manipulation d'entités, issues d'une description par régions.

Comment peut-on définir une stratégie pour passer d'une représentation sous forme de signal numérique à une représentation symbolique d'une scène? Ce problème très général d'analyse de scène peut être résolu de manière satisfaisante lorsque l'on connaît la scène à analyser donc les objets qu'elle contient. Le problème devient beaucoup plus complexe si l'on n'a aucune connaissance a priori de ce que cette scène représente. Le processus de reconnaissance par une machine ou un être humain nécessite un apprentissage, la caractérisation du monde à interpréter au moyen de règles ou de termes connus. La modélisation implique toute une hiérarchie de niveaux de traitement où les symboles sont manipulés selon des règles d'inférence. Il s'agit de définir ces symboles, c'est-à-dire un alphabet simple permettant de caractériser des groupes de symboles d'un niveau d'abstraction (hiérarchique) inférieur.

Pour obtenir une segmentation, on a cherché à modéliser mathématiquement ou statistiquement les variations spatiales de luminance. On désire extraire des paramètres de l'image pouvant définir des invariants. A partir de là, il est associé à chaque région une série d'attributs. Ceux-ci correspondent souvent, du moins en partie, à des entités ayant une signification sémantique, telles des caractéristiques de surface d'un objet. L'interprétation se fait par interaction entre ces différents attributs selon un modèle de connaissance. Le processus d'interprétation permet le plus souvent de parfaire la partition de l'image.

S'il est possible de faciliter l'interprétation d'une image à partir d'une représentation segmentée de celle-ci, une segmentation permet aussi de réduire

considérablement la redondance d'information présente dans une image. Son utilisation dans le contexte du codage d'images est donc souhaitée. En vue de cette application, il est important de souligner les critères nécessaires à une compression élevée sans perte sensible d'information sémantique. En liaison avec la représentation segmentée de l'image, il faut :

- reproduire le plus fidèlement possible les contours réels des objets, c'est-à-dire décrire les frontières des régions qui leur correspondent avec une grande précision.

- admettre une certaine tolérance dans la représentation du signal image à l'intérieur de chaque région.

- minimiser le nombre de régions de manière à éviter de coder des frontières artificielles et un nombre de paramètres d'approximation trop élevé. A chaque région est associée une certaine approximation.

- éventuellement ajouter une composante aléatoire à l'image codée après segmentation de façon à pouvoir lui redonner un aspect plus naturel.

Voyons donc les raisons de ces exigences. On va montrer la nécessité de *préserver l'information contour* en liaison avec les propriétés du système visuel humain. Un modèle possible du système visuel humain est présenté à la figure 2.5 [1].

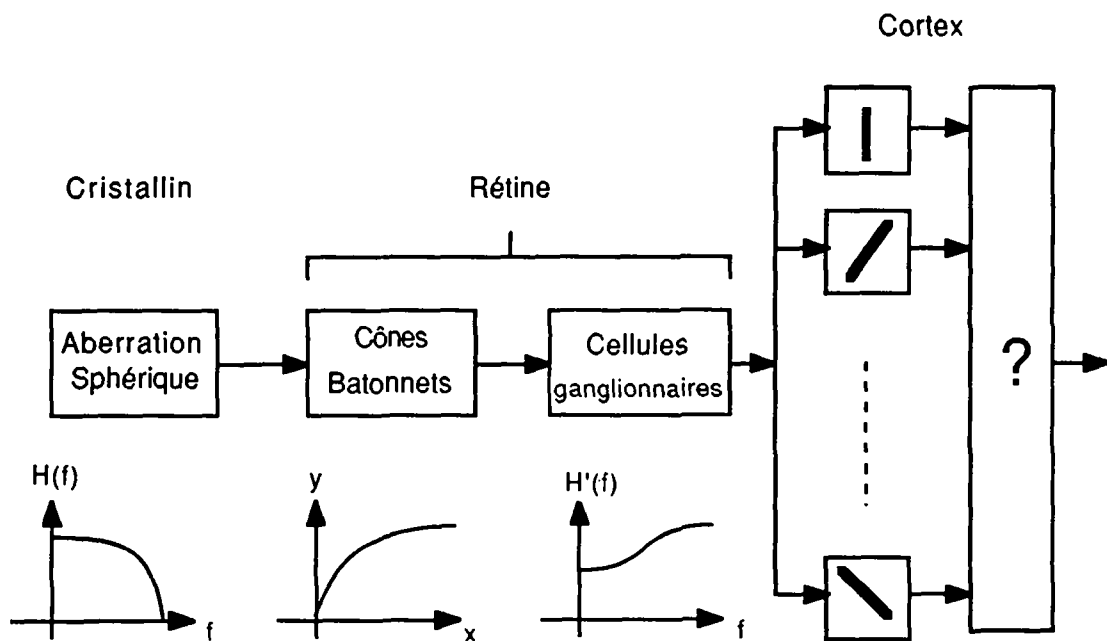


Figure 2.5: Principe de fonctionnement du système visuel humain

Dans un premier temps, on peut considérer en première approximation que l'image est filtrée passe-bas de manière isotropique. Le phénomène physique à la base de ce filtrage correspond à l'aberration sphérique du cristallin. La sensibilité des photorécepteurs (cônes et bâtonnets) au signal image se formant sur la rétine est de type logarithmique. Enfin, le processus d'inhibition latérale des cellules

ganglionnaires agit comme un filtre passe-haut mettant en évidence l'information contour. Au niveau du cortex, cette information est séparée selon différentes directions. Ce modèle très simplifié peut être élaboré à partir des travaux des deux neurobiologistes Hubel et Wiesel [30]. Il permet de comprendre l'importance du contenu sémantique associé aux contours, c'est-à-dire les endroits de variation brusque du signal image. Sans aller jusqu'à analyser la structure même du système visuel humain, on peut se rendre compte de ce phénomène en comparant une scène et l'image des contours associée (Voir figure 2.6). L'information sémantique liée à la scène est bien présente au niveau de l'image des contours.

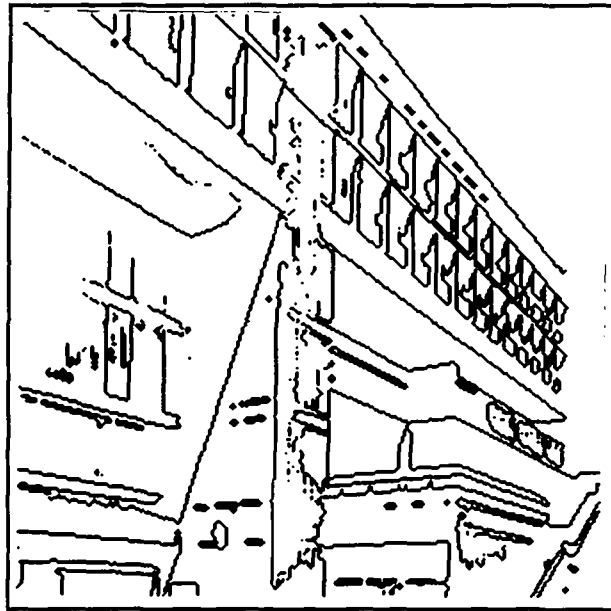


Figure 2.6 : Image des contours du bâtiment

De manière à limiter la dégradation d'une image quant à son contenu sémantique, il faut absolument préserver l'information contour après segmentation, autrement dit :

- éviter de perdre des contours de l'image originale,
- ne pas insérer de contours artificiels, c'est-à-dire inexistant dans l'image originale.

Le non-respect de cette règle résulte le plus souvent en une distorsion parfois irréversible pour l'interprétation. Il est difficile de mesurer objectivement la dégradation subie lors de la modification de l'information contour car son effet est très dépendant de la structure de la scène analysée. On peut néanmoins dire que des contours isolés sans "régularité" correspondent le plus souvent à des composantes aléatoires du signal et sont donc peu porteurs d'information sémantique.

Une zone est dite homogène si tous ses éléments vérifient une même propriété définie à l'avance. La segmentation est l'opération qui permet d'extraire les régions homogènes de l'image. Bien entendu, les frontières de ces régions, si elles correspondent à des contours réels de l'image, doivent être décrites avec la plus grande précision de manière à épouser ceux-ci. Aucune approximation de

frontière ne saurait être tolérée sans avoir analysé la dégradation qui en résulterait au niveau du contenu de l'image.

Quant aux raisons de la deuxième exigence, c'est-à-dire l'approximation des données sur la région, il faut réaliser qu'une compression importante de l'information ne peut être obtenue qu'avec une approximation. Comme celle-ci est interdite au niveau des frontières, elle ne peut intervenir qu'au niveau des données images.

La troisième exigence est liée à l'importance de la compression. En effet, la représentation de chaque région implique la description de ses frontières et de l'approximation du signal luminance qu'elle renferme.

La quatrième exigence se justifie à cause de l'aspect souvent très basse fréquence des régions de grande taille où le modèle d'approximation ne permet pas de reproduire certaines granularités existant dans le signal original.

Commentaires:

1) L'approche de Riseman et Arbib [31] qui consiste à obtenir une segmentation d'une image pour en faciliter son interprétation est controversée. En effet, ne faut-il pas tout de suite essayer d'utiliser la sémantique pour obtenir une bonne segmentation? Le contexte local dans une image influence directement l'interprétation qui en est faite [32]. Tenenbaum et al. [33] reprochent à une modélisation mathématique ou statistique du signal luminance seul, de présenter des limitations considérables. Ceci, car ce genre de caractérisation ne tient absolument pas compte de la structure physique de la scène qui est analysée ni de la manière dont a été formée l'image. Ils opposent ainsi une représentation de l'image obtenue à partir d'un "modèle structurel" (qui tient compte des réalités physiques de la scène) à celle obtenue à partir d'un "modèle statistique". D'après eux, une approche statistique ne se justifie que pour modéliser la composante aléatoire du signal, c'est-à-dire le bruit ou les textures aléatoires. Par contre une approche structurelle s'impose en relation avec l'attitude de l'être humain lorsqu'il regarde une scène. Ce qu'un homme voit au premier coup d'oeil, ce sont les caractéristiques physiques des surfaces des objets qui lui sont présentées indépendamment du point de vue choisi ou de la position de la source lumineuse.

Nous pensons pour notre part que les autres approches ont leur raison d'être soit parce qu'elles s'appliquent à un contexte spécifique où des études simples des caractéristiques du signal luminance correspondent à une réalité physique sous-jacente (ex: détection d'objets plans dans de bonnes conditions d'éclairage), soit parce que la définition d'un modèle de connaissance général d'un système de vision est très difficile à mettre en oeuvre sans utiliser une représentation symbolique justement donnée par la segmentation.

2) Le signal image est décrit pour chaque région de manière relativement simple au moyen d'une approximation. L'aspect modélisation du signal luminance est donc conservé tout en insérant des considérations structurelles par une approche adaptative pour aboutir à la segmentation. Le modèle permettant de décrire le signal luminance est suffisamment souple pour s'adapter aux caractéristiques locales de l'image.

L'aspect structure de la méthode présentée réside essentiellement dans la parfaite adéquation entre les caractéristiques physiques de la scène analysée et l'approximation qui est utilisée pour les représenter. Tout se passe comme si à partir d'un certain modèle du monde, on cherche à trouver la meilleure représentation d'une scène analysée avec ce modèle. Parmi toutes les configurations possibles de segmentation ayant un nombre donné de régions, il s'agit de définir comment obtenir la meilleure correspondance entre les données images et l'approximation, en jouant sur les paramètres de celle-ci et la forme des régions.

3) La propriété intervenant dans la définition d'une zone homogène joue un rôle très important. Pour qu'elle nous permette d'atteindre nos buts, il faut qu'elle traduise fidèlement l'impression visuelle de régularité et de continuité. Dans ce cas la segmentation ne peut aboutir qu'à la partition de l'image où les frontières des régions épousent les contours des objets.

2.1.2 Approche adaptative :

Par définition, un système ou une méthode sont adaptatifs si leur comportement peut être modifié en fonction de l'entrée.

Les données provenant de la numérisation d'images naturelles sont très inhomogènes. Par exemple, dans un portrait, les cheveux n'ont pas le même degré de détail que la joue ou le front. Une méthode conçue pour traiter le détail fin des cheveux ne donnera aucun résultat sur la joue ou le front si on ne peut modifier son comportement.

Toute méthode de traitement d'images naturelles se doit donc d'être adaptative si l'on souhaite des bonnes performances et une efficacité accrue.

Nous examinons ci-dessous comment l'adaptivité peut intervenir dans la segmentation.

Il existe deux approches possibles au problème de segmentation :

- la formation de frontières : cette approche consiste à trouver les frontières qui délimitent une région (approche directe),
- la formation de régions : il s'agit ici d'analyser des caractéristiques de certaines zones de l'image de manière à les découper ou à les rassembler en régions (approche indirecte).

L'attention est portée dans le premier cas sur la localisation des différences de propriétés alors que dans le second, on cherche à mettre l'accent sur les similitudes de propriétés. Il est possible que ces deux approches donnent des résultats comparables. Selon le but de la segmentation et le type de données images, il est plus simple d'utiliser l'une ou l'autre de ces méthodes.

Ainsi posé, le problème de la segmentation devient très proche de celui de l'extraction de contours. Pour les raisons déjà citées, nous adopterons l'approche indirecte. Dans ce cas, le problème apparaît plus dans la définition du modèle permettant de décrire des propriétés de zones de l'image en vue de sa partition

ultérieure. Il faut également choisir des critères pour évaluer la représentativité de ce modèle par rapport aux données de départ. En effet, une approximation bien adaptée à une certaine zone de l'image peut se révéler totalement insuffisante lorsqu'on considère une zone plus étendue. Cette considération nous amène tout naturellement à reprendre la notion de zone homogène qu'on appelle parfois "espace régulier". Cette idée fut utilisée par M. Kocher [27], puis reprise par J.C. Simon [34].

L'adaptation dans la segmentation peut intervenir soit au niveau des paramètres du modèle, soit sur le domaine où le modèle est utilisé. Dans le premier cas, les valeurs des paramètres sont modifiées de manière à vérifier l'homogénéité de la région analysée. Dans le deuxième cas, la forme de la région est modifiée pour effectuer le même test. Si aucun de ces tests ou de leurs combinaisons n'aboutit à un résultat positif, alors la frontière de la zone homogène (espace régulier) est atteinte. A partir d'un nombre restreint de pixels judicieusement choisis, il est possible d'utiliser un certain modèle pour caractériser une propriété d'une bien plus grande région de l'image. L'analyse de cette propriété pour des configurations similaires de pixels ne doit permettre que de renforcer (rendre plus robuste) le modèle.

2.1.3 Adaptation des paramètres du modèle :

Le modèle choisi pour caractériser une zone de l'image doit être simple. Il doit en outre définir le comportement du signal luminance sur cette zone. Nous allons admettre qu'à partir d'un modèle, il est possible de définir une approximation du signal original que nous noterons $g^{\wedge}(x,y)$. Pour pouvoir adapter $g^{\wedge}(x,y)$ aux données images sur un certain domaine D , il faut rendre $g^{\wedge}(x,y)$ dépendant d'un certain nombre de paramètres dont la valeur sera fixée par le choix d'un critère d'approximation du signal image.

En jouant sur tous les paramètres de l'approximation, on cherche à minimiser une certaine fonction d'erreur \mathbb{A} entre le signal original et son approximation sur le domaine D analysé. Cette fonction définit le critère d'approximation (voir section 2.3). En adaptant le modèle, c'est-à-dire les paramètres de la fonction $g^{\wedge}(x,y)$, aux caractéristiques du signal du domaine D , on peut extraire de bons attributs permettant de caractériser cette région de l'image. A partir d'un ensemble de primitives, par exemple, des polynômes bidimensionnels, on cherche quel jeu de paramètres de position, de pente, de courbure... permet de mieux représenter le signal image sur un domaine donné. On transpose une réalité complexe sur un nombre réduit de ces primitives, et ceci pour chaque objet ou partie d'objet de l'image.

2.1.4 Adaptation de la forme des régions : Division et Rassemblement :

Il s'agit ici de modifier la forme et la taille de chaque domaine D de manière à garantir que \mathbb{A} ne dépasse pas un certain seuil.

Dans un premier temps, l'image est subdivisée de façon adaptative en régions de différentes tailles. Chaque région constitue une partie homogène du signal luminance que l'on est capable de reproduire presque parfaitement au moyen de notre ensemble de primitives, c'est-à-dire les paramètres de

l'approximation choisie de manière à minimiser la fonction d'erreur \mathbb{A} . Les zones ne pouvant pas être considérées comme régulières se limitent à la taille d'un pixel. Cette première phase de la segmentation appelée *division adaptative* permet ainsi d'obtenir des parties d'espaces réguliers, c'est-à-dire des parties d'objets de l'image originale. Elle fait l'objet du chapitre 3.

Supposons donc avoir obtenu cette partition initiale. Pour les régions trouvées, on est sûr que $g^{\wedge}(x,y)$ approxime fidèlement le signal image, le seuil de qualité n'ayant pas été franchi. Il s'agit maintenant de regrouper certaines régions (voire inclure des points isolés à des régions), de manière à ce que la segmentation finale corresponde aux caractéristiques physiques de la scène. Ce regroupement s'avère d'ailleurs d'autant plus nécessaire vu qu'il faut limiter autant que possible le nombre de régions pour augmenter la compression d'information. On décide ainsi de réunir deux à deux toutes les régions adjacentes qui possèdent des caractéristiques similaires.

Deux régions sont considérées comme adjacentes ou contiguës selon le principe de connexité d'ordre 4. On a préféré cette mesure de voisinage à une mesure de connexité d'ordre 8 pour deux raisons. On réduit d'une part le nombre de voisins possibles. On assure d'autre part sur le plan de l'appréciation visuelle la continuité du signal luminance sur une longueur finie (au moins la largeur du domaine d'intégration de la lumière correspondant à un pixel). En effet, en utilisant une connexité d'ordre 8, l'homogénéité de la zone peut être transmise par un point P sans dimension comme illustré à la figure 2.7.

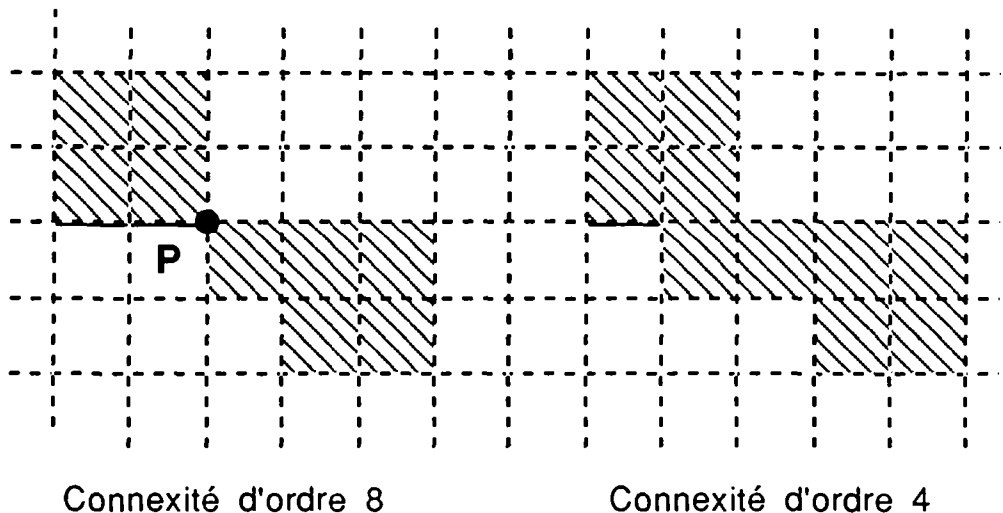


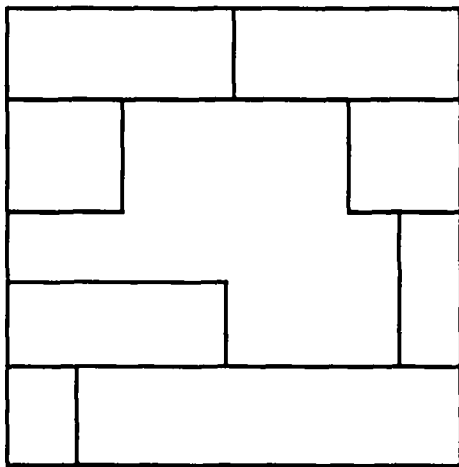
Figure 2.7: Continuité d'un domaine D connexe

- a) Connexité d'ordre 8: la continuité est assurée par un point P .
- b) Connexité d'ordre 4: la continuité est assurée sur au moins la largeur du domaine correspondant à un pixel.

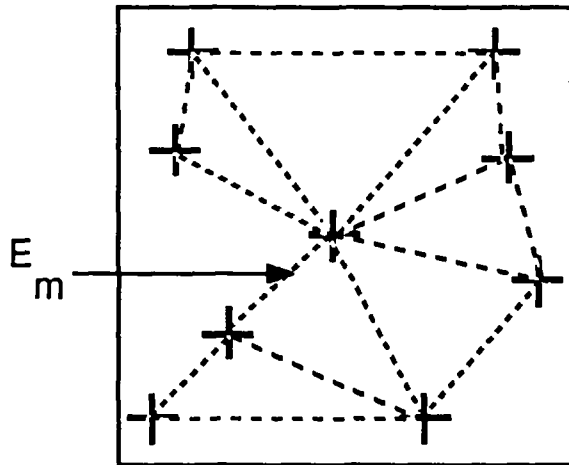
Deux domaines adjacents sont rassemblés chaque fois qu'ils correspondent à des parties d'un même espace régulier. La mesure de l'homogénéité des caractéristiques doit se faire en relation avec le changement des fonctions d'approximation $g^{\wedge}(x,y)$ sur l'ensemble des deux domaines. Moins il y a de modification, plus on peut s'attendre à ce que ces deux domaines définissent un seul et même espace régulier. Le problème reste qu'il faut placer un seuil sur la mesure d'homogénéité avant d'arrêter le processus de rassemblement. L'étude des méthodes de segmentation basées sur ce principe [27] a montré qu'au-delà de

certaines valeurs, il était illusoire de réduire le nombre de régions. On aboutit soit à une image fortement dégradée (l'information sémantique n'est pas préservée), soit à un nombre excessif de régions. On remarque en outre une forte dépendance du choix initial de la première zone à regrouper. De manière à se rendre indépendant de l'endroit où l'on commence la croissance, Kocher et al. [35] ainsi que Morris et al. [36] ont ensuite développé grâce à la théorie des graphes une technique adaptative de croissance de région.

L'idée consiste à chercher l'endroit où l'association de deux zones de l'image entraîne la plus faible dégradation. Le processus est itéré jusqu'à obtenir une segmentation jugée satisfaisante. Prenons la partition d'une image présentée à la figure 2.8a. On passe d'abord à une représentation par graphe de contiguïté de région [37]. (GCR). Chaque région constitue un noeud du graphe; chaque branche définit deux régions adjacentes. A partir du GCR, il est possible de remonter sans équivoque à l'information de voisinage de chaque région. La figure 2.8b représente le GCR associé à la figure 2.8a.



a) Partition d'une image

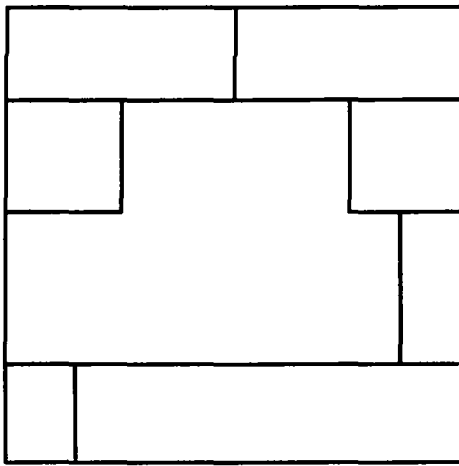


b) Graphe de contiguïté de région (GCR) associé

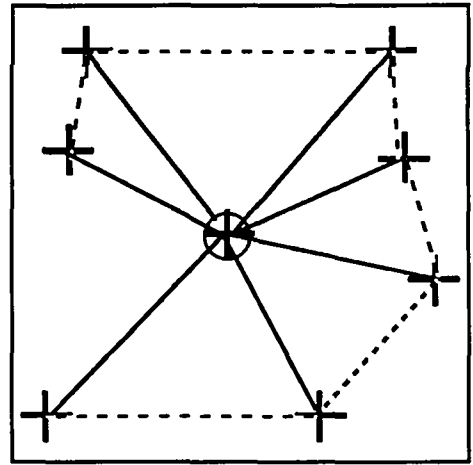
$$E_m = \text{minimum de dissimilarité}$$

Figure 2.8.

Dans un deuxième temps, on associe à chaque branche du noeud un coût qui indique le degré de dissimilarité entre les deux noeuds. En cherchant le coût le plus faible sur l'ensemble du GCR, il est possible de trouver les deux régions voisines ayant les caractéristiques les plus proches, régions que l'on peut dès lors regrouper. Après chaque rassemblement, le GCR est modifié. Les noeuds des régions où l'on a effectué le rassemblement sont associés en un seul noeud, noeud qui est relié à l'ensemble des voisins de la région ainsi constituée. Après un premier rassemblement, le second s'effectuera peut-être à un autre endroit du GCR, là où la similarité des deux régions sera jugée la plus grande. Les figures 2.9a et 2.9b donnent la partition de l'image et le GCR correspondant après une étape de regroupement en partant de la partition selon la figure 2.8a.



a) Partition



b) GCR

⊕ : Nouveau noeud

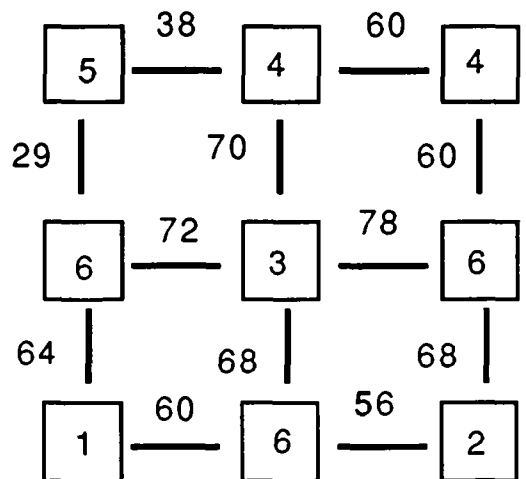
— Nouvelles branches

Figure 2.9: Première étape de rassemblement à partir de la segmentation de la figure 2.8.

La figure 2.10 présente deux étapes de regroupement lorsqu'on a une partition initiale de 9 carrés avec 4 pixels chacun. Pour chaque région, la dissimilarité est mesurée en estimant l'erreur quadratique si les régions sont approximées par leur valeur moyenne.

6	3	8	5	5	8
6	4	1	2	2	1
8	7	8	3	8	7
5	2	1	0	7	2
2	1	4	9	5	1
1	0	5	4	0	2

a) Image originale
(niveaux de gris)



b) Partition initiale

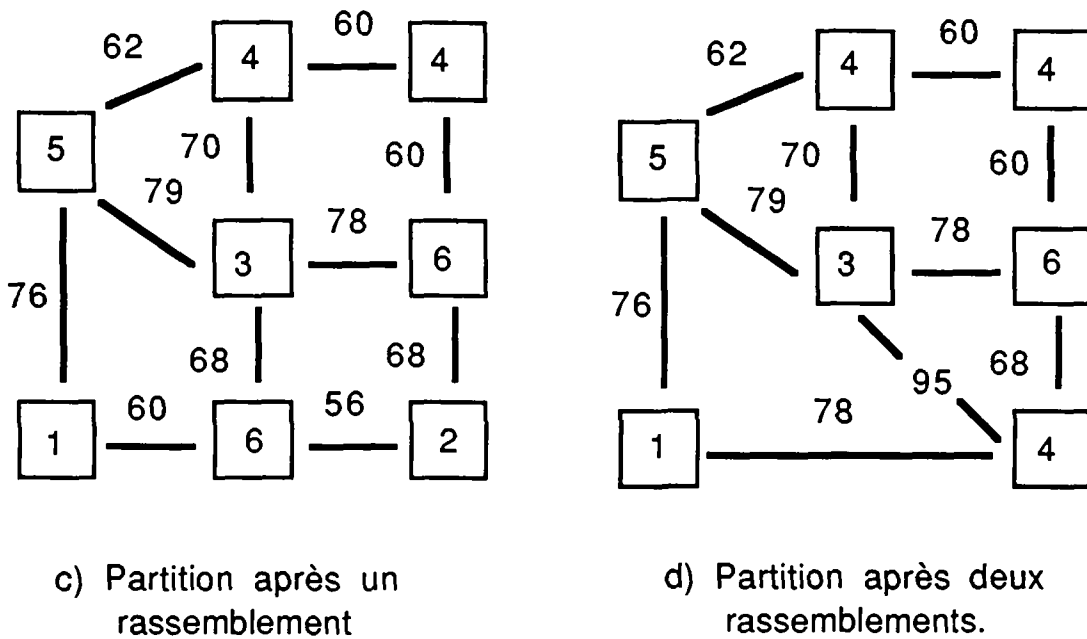


Figure 2.10: Exemple de rassemblement adaptatif. Partition initiale de 9 carrés de 2x2 pixels. A chaque noeud du GCR est associée la valeur moyenne de la région associée. A chaque branche est associée l'erreur quadratique qui résulterait après moyennage des niveaux de gris de l'association des deux noeuds que cette branche relie.

Le regroupement s'arrête lorsqu'on atteint un nombre de régions donné ou lorsqu'on commence à dégrader trop fortement l'image. Cette méthode est utilisée pour modifier la segmentation initiale (après division) par rassemblement de régions. Elle fait l'objet du chapitre 4.

La démarche de division et rassemblement est similaire à celle d'un être humain. A partir d'une appréciation globale de la scène, on descend à une analyse locale pour remonter enfin aux objets de l'image.

Une fois la segmentation obtenue, on pourra représenter l'information en décrivant la position des régions ainsi que les paramètres d'approximation judicieusement quantifiés relatifs à chacune de ces régions. A partir de ces deux informations, il sera possible de reconstruire le signal luminance approximé pour chaque région, donc chaque objet ou partie d'objet de l'image.

Si le principe de division et rassemblement que l'on vient de décrire a été proposé par Horowitz et Pavlidis [25], on l'a rendu adaptatif. On a défini aussi une manière de modéliser de façon optimale les caractéristiques du signal image, à chaque étape de la segmentation.

2.1.5 Complexité du problème :

La transformation des données images par minimisation de la fonction d'erreur A nécessite la résolution d'un système de r équations à r inconnues où r constitue le nombre de paramètres d'approximation (voir section 2.3).

Aussi bien durant la partition initiale de l'image qu'au cours du rassemblement adaptatif, il faut effectuer ce calcul chaque fois que l'on désire modifier la forme d'un domaine.

Pour chaque domaine D à regrouper à l'un de ses voisins, il faut évaluer le degré de similarité qu'il a avec chacun d'entre eux. Ensuite, l'on compare ces valeurs avec toutes celles qui sont associées aux autres branches du GCR. On a réduit la complexité du problème en limitant le nombre de régions de la segmentation avant rassemblement adaptatif par l'application de la stratégie adaptative de division. Dans [35], Kocher et al. proposent de réduire ce nombre en décomposant l'image en des carrés de même taille. Celle-ci est fixée en relation avec le nombre minimum de pixels qu'il faut considérer pour permettre leur approximation optimale au sens des moindres carrés par un polynôme bidimensionnel. Ainsi, lorsqu'on utilise un polynôme permettant de décrire une conique ($r=6$), il faut diviser l'image en un ensemble de carrés 3×3 . Cette méthode présente néanmoins deux désavantages:

- d'une part, on ne peut pas atteindre une précision extrême dans la description des frontières des régions. En effet, la méthode telle qu'elle est décrite ci-dessus limite la résolution au nombre de points se trouvant sur le côté des carrés initiaux (Dans l'exemple précédent, la précision serait réduite à 3 points). Or, ceci ne correspond en général pas à la réalité physique de l'image. A proximité de la frontière entre deux objets, il aurait fallu décomposer ultérieurement les carrés en leurs points constituants, ceux-ci ne pouvant définir de zone homogène.

- d'autre part, on subdivise trop l'image originale. Il est possible qu'il existe des zones beaucoup plus grandes que ces carrés, zones qui correspondent déjà à des parties d'espaces réguliers. Ces régions sont obtenues dans notre cas en imposant un seuil à ne pas franchir à la fonction d'erreur \mathbb{A} . Le rassemblement adaptatif par croissance de région selon [35] ou après division aboutit au même résultat si ce n'est qu'on a limité dans le deuxième cas le nombre de régions à rassembler.

Pour diminuer la complexité du problème sans aller changer la stratégie adaptative de rassemblement pratiquée sur l'ensemble de l'image, il suffit de réduire le nombre de paramètres d'approximation qui permettent de décrire chaque région. D'un autre côté, augmenter sensiblement le nombre de paramètres du modèle a l'avantage de limiter le nombre de régions définissant des zones homogènes, vu que l'on est à même de décrire des caractéristiques plus complexes du signal luminance.

De manière à faciliter les opérations de division et rassemblement, il apparaît essentiel d'extraire des points de repère sur l'image à traiter. C'est ce que nous nous proposons d'étudier dans ce qui suit.

2.2 Extraction d'images de contrôle :

Pour piloter le processus de segmentation, on va utiliser des images de contrôle. Celles-ci repèrent des points particuliers de l'image analysée, points correspondant à un contenu sémantique important. De ce qui a été présenté au paragraphe 2.1.1, l'information contour est primordiale dans une image. Les images de contrôle permettent de définir les contours réels des objets que certaines frontières des régions doivent épouser sans distorsion. A partir d'une image de contrôle des contours, l'erreur d'approximation entre le signal original et sa représentation $g^{\wedge}(x,y)$ peut être mesurée au voisinage de ces contours. Toute valeur excessive de cette erreur entraîne une modification de la segmentation. Il est également possible de détecter l'apparition de faux contours dans l'image approximée. Ceux-ci peuvent alors être éliminés soit en changeant la segmentation, soit par l'application locale d'un filtre passe-bas de rehaussement (voir chap. 5).

2.2.1 Hypothèses sur les images traitées:

Dans ce travail, on fait l'hypothèse que les surfaces des objets que contiennent les images naturelles traitées ont une luminance à variation lente. Définir les frontières réelles des objets correspond alors simplement à extraire les contours de l'image, où ceux-ci définissent des changements brusques du signal. Pour réduire la complexité d'une scène, il faut réduire autant que possible le nombre d'objets qu'elle contient, c'est-à-dire le nombre de régions de la segmentation. Ceci peut être réalisé en utilisant des modèles plus élaborés pour décrire chaque région. Il nous a paru qu'il était plus simple de caractériser dans un premier temps des régions à variation lente qui représentent en général la majeure partie d'une image naturelle.

2.2.2 Extraction de contours:

Cette étape préliminaire à la segmentation permet d'extraire une image binaire de contrôle $c(k,l)$ qui permet de localiser les contours de la scène. Ce paragraphe ne propose pas de trouver le remède miracle à ce problème qui a déjà été traité en détail mais justifie certains choix en liaison avec les buts que l'on s'est fixés. Comme on a vu que les contours se définissent par les endroits de variation rapide de luminance, on peut les détecter en estimant des dérivées du signal image. On ramène ainsi le problème d'extraction de contours à un problème de différentiation numérique. A cause de la sensibilité au bruit de la dérivation, il s'agit de régulariser l'image avant d'en calculer la dérivée [38]. Cette régularisation permet de bien poser le problème de la détection de contours au sens de Hadamard, c'est-à-dire:

- 1.- sa solution existe,
- 2.- elle est unique,
- 3.- elle dépend continuellement des données de départ.

Parmi les différentes méthodes permettant d'atteindre cette régularisation, on peut utiliser des opérateurs de stabilisation correspondant à un filtrage passe-bas.

Ceci introduit naturellement une perte de l'information haute fréquence du signal, ce qui entraîne souvent une mauvaise localisation des contours. La fonction de filtrage donnant le meilleur compromis théorique est la fonction gaussienne [38]. C'est ce qui explique en partie les performances des opérateurs de Marr-Hildreth [39] ou de Canny [40]. Dans le même ordre d'idées, Hueckel [41] a proposé d'approximer au sens des moindres carrés le signal luminance sur une région de l'image au moyen de fonctions orthogonales. Plus récemment, Haralick [42] justifia l'utilisation de fonctions polynomiales pour approximer le signal luminance de manière à augmenter la robustesse de l'extraction. D'autres techniques de régularisation peuvent être envisagées en appliquant à l'image des filtres non linéaires de lissage qui sont supposés préserver l'information contour mais éliminent la granularité locale (filtrage médian [43], filtre de Nagao [44]). Chaque fois, il est apparu que ces filtres provoquent une perte de certains détails importants tels points anguleux ou lignes contrastées. Les points anguleux correspondent aux endroits où se croisent deux contours; les lignes contrastées définissant des contours très étroits sur un fond homogène.

La figure 2.11 montre le résultat de filtrage par une gaussienne d'écart types 0.117 et 0.176 pour l'image du bâtiment et 0.059 et 0.117 pour l'image du couple dans le domaine fréquentiel (Les valeurs des écarts types sont indiquées par rapport à une fréquence normalisée à 0.5). La figure 2.12 présente l'application du filtre de Nagao après une et deux itérations pour les mêmes images de départ. Enfin, la figure 2.13 donne ces images filtrées par l'opérateur médian en utilisant des fenêtres 3x3 et 5x5. On remarque que les déformations sont souvent introduites aux points anguleux. En outre, le filtre de Nagao crèent de faux contours (visibles surtout sur le front des personnages du couple) alors que le filtrage médian tend à lisser la forme des contours (visible surtout sur le bâtiment).



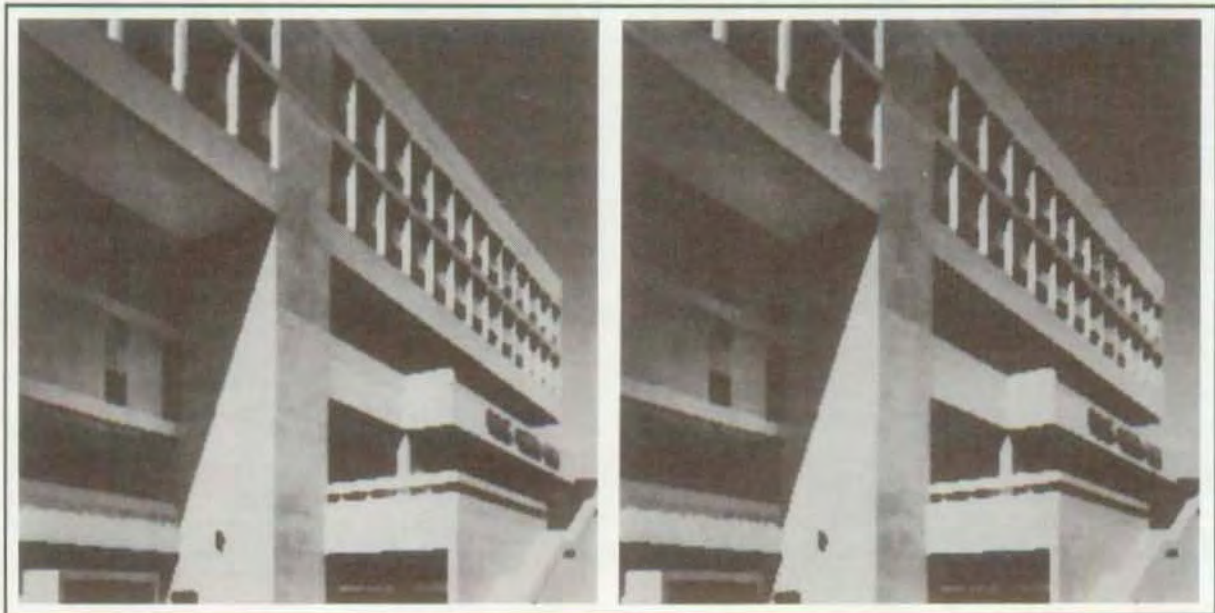
a) Bâtiment ($\sigma=0.117$)

b) Bâtiment ($\sigma=0.176$)

c) Couple ($\sigma=0.059$)d) Couple ($\sigma=0.117$)

Figure 2.11: Résultats de régularisation par divers filtres passes bas de réponse fréquentielle gaussienne

$$H_{\text{gauss}}(\rho, \theta) = \exp(-\rho^2/\sigma^2)$$



a) Bâtiment (une itération)

b) Bâtiment (deux itérations)



c) Couple (une itération)

d) Couple (deux itérations)

Figure 2.12: Résultats de régularisation par filtrage de Nagao 5x5 selon [44].



a) Bâtiment (fenêtre 3x3)

b) Bâtiment (fenêtre 5x5)



c) Couple (fenêtre 3x3)

d) Couple (fenêtre 5x5)

Figure 2.13: Résultats de régularisation par filtrage médian

Les contours sont détectés à partir d'une dérivée du signal régularisé. L'application d'opérateurs qui estiment la dérivée première du signal tels celui de Roberts [14] ou de Sobel [45], nécessite la recherche de maxima dans le signal filtré. Il s'agit de trouver également un moyen d'éliminer certains maxima car le signal filtré peut présenter plusieurs maxima voisins correspondant à un seul contour réel de l'image. Pour cette raison, l'utilisation d'opérateurs permettant d'estimer la dérivée seconde du signal est devenue plus courante. Les contours correspondent alors simplement aux points d'inflexion du signal filtré. Comme celui-ci est en principe à valeur moyenne nulle, ces points d'inflexion correspondront aux passages par zéro. Certains de ces opérateurs tels le Laplacien sont invariants par rotation. Il faut éviter la création de passages par zéro secondaires correspondant à des phénomènes de Gibbs. En plaçant un seuil sur l'amplitude du signal différencié, c'est-à-dire la différence entre le minimum et le maximum apparaissant de part et d'autre d'un passage par zéro, on peut éliminer bon nombre de passages par zéro secondaires. Un autre moyen consiste à élargir la largeur de la bande de transition des caractéristiques fréquentielles du filtre de différenciation. Il ne faut pas trop excéder dans ce sens sans quoi on risque d'avoir une mauvaise localisation des contours réels.

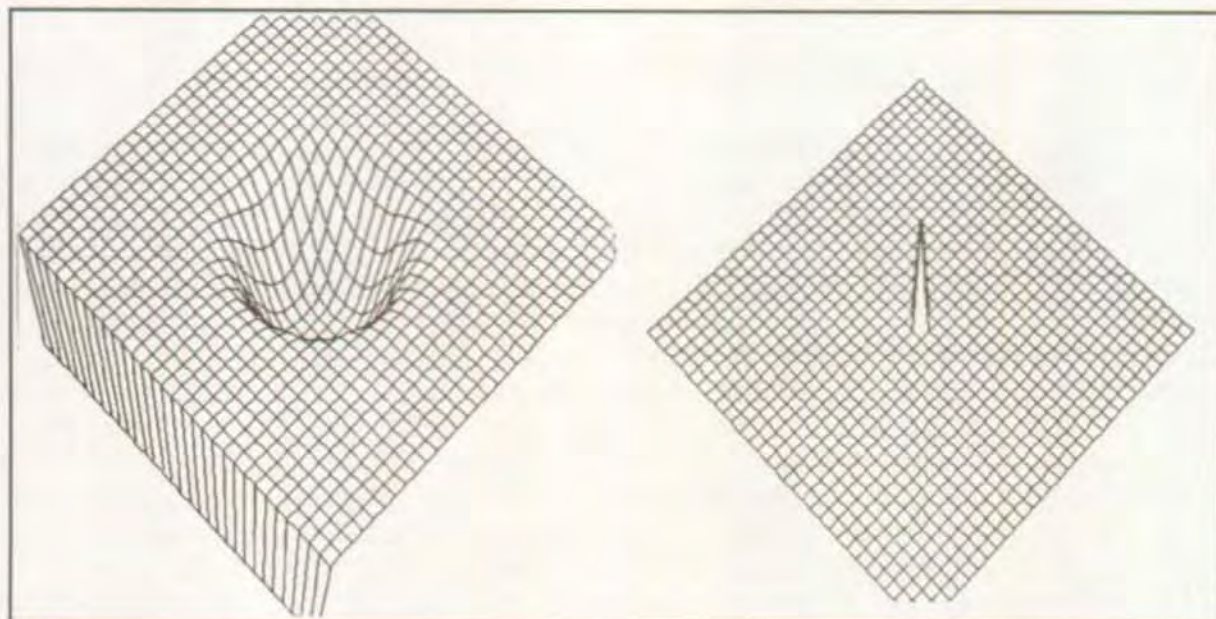
Une façon simple d'effectuer l'estimation de la dérivée seconde est obtenue à l'aide d'un filtre passe-haut idéal que l'on multiplie par une fenêtre gaussienne dans le domaine spatial pour avoir une réponse impulsionnelle finie. La convolution qui en résulte dans le domaine fréquentiel donne comme résultat une fonction erf centrée à la fréquence de coupure du filtre passe-haut idéal. Le filtre haute fréquence obtenu a donc pour réponse fréquentielle:

$$\begin{aligned}
 H(\rho, \theta) &= \{1/(\sigma\sqrt{2\pi})\} \cdot \int_{-\infty}^{\rho} \exp(-((\rho'-f_c)/(\sigma\sqrt{2}))^2) d\rho' & \text{si } \rho > 0 \\
 &= 0 & \text{si } \rho = 0
 \end{aligned}
 \tag{2.1}$$

où ρ et θ définissent un système de coordonnées fréquentielles polaires. Elles sont reliées aux fréquences spatiales u et v par les relations :

$$\begin{aligned} u &= \rho \cos\theta \\ v &= \rho \sin\theta \end{aligned} \quad (2.2)$$

Les paramètres σ et f_c de ce filtre permettent de s'adapter aux caractéristiques de détection de contours [8]. f_c est fixé relativement bas pour préserver la majeure partie de l'information haute-fréquence. σ doit être fixé proche de f_c pour éviter l'apparition de passages par zéro secondaires; il ne faut cependant pas lui donner une valeur supérieure à $3/4f_c$ sans quoi on risque de ne pas atténuer suffisamment les composantes très basses fréquences de l'image rendant la détection des passages par zéro erronée. Dans les exemples qui suivent $f_c=0.027$ $\sigma=0.02$ par rapport à une fréquence normalisée à 0.5. La figure 2.14 montre les réponses fréquentielle et impulsionnelle de ce filtre.



a) Réponse fréquentielle

b) Réponse impulsionnelle.

Figure 2.14: ($f_c=0.027, \sigma=0.02$)

De manière à améliorer la détection, on peut tenir compte du contexte par poursuite de contours. Les seuils sur l'amplitude des transitions aux passages par zéro sont modulés en fonction de l'existence de contours voisins déjà détectés. En partant ainsi d'un maximum de transition (contour très marqué), on recherche les chemins que définissent tous les passages par zéro liés à ce point. Ceux-ci définiront des contours de l'image si l'amplitude des transitions estimée perpendiculairement à la direction du chemin reste supérieure à un seuil de fin de poursuite. L'algorithme continue tant qu'il reste des maxima de transition dépassant un seuil de début de poursuite. Le seuil de début de poursuite est situé beaucoup plus haut que l'autre seuil, de manière à commencer toujours sur un

point de contour important. La figure 2.15 présente à cet effet l'image du couple après application du filtre de réponse fréquentielle (2.1). Plus précisément, la figure a) montre l'image filtrée, la figure b) la position des passages par zéro, la figure c) l'amplitude des transitions en ces points et enfin la figure d) les contours détectés.



a) Image filtrée

b) Position des passages par zéro de a)

c) Amplitude des transitions
aux passages par zérod) Contours détectés
(Seuil de début de poursuite=5.8%)
(Seuil de fin de poursuite =11.6%).

Figure 2.15: Détection des contours de l'image du couple en utilisant l'opérateur $H(\rho, \theta)$

Une question intéressante consiste à savoir s'il existe un opérateur de différentiation proche du traitement qu'effectue le système visuel humain dans la détection des contours. La physiologie de l'oeil nous apprend que certaines cellules ganglionnaires de la rétine effectuent un filtrage similaire au Laplacien d'une gaussienne, ainsi que Marr et Hildreth l'ont proposé [39]. Marr et Poggio [46] ainsi que Ikonomopoulos et Kunt [7] ont suggéré l'utilisation d'opérateurs de dérivation directionnelle en relation avec l'organisation du cortex visuel. Il est cependant difficile de comparer sur le plan psychovisuel la qualité de détection de ces opérateurs sans définir des bancs de tests exhaustifs [38].

Nous allons dans un premier temps discuter les résultats après le filtrage de Marr-Hildreth dans le contexte de notre application. La figure 2.16 présente la détection obtenue à partir de l'image du couple pour différentes valeurs de l'écart type et du seuil de détection sur l'amplitude du profil du signal à un passage par zéro. On remarque que plus le premier paramètre décroît plus on régularise l'image tout en dégradant cependant la position des contours. Un bon compromis semble possible en choisissant un écart type de 0.117 et un seuil de détection de 5% de la dynamique des amplitudes des passages par zéro. La plupart des contours réels sont ainsi préservés et il ne reste que peu de granularité. Idéalement, le seuil de détection devrait être fixé de manière locale en tenant compte de l'histogramme des amplitudes de l'image (une région contrastée aura ainsi son seuil plus élevé).



a) $\sigma=0.058$; $T=1.9\%$

b) $\sigma=0.117$; $T=1.9\%$



c) $\sigma=0.136$; $T=7.8\%$

d) $\sigma=0.117$; $T=7.8\%$

Figure 2.16: Détection des contours de l'image du couple en utilisant l'opérateur de Marr-Hildreth

$$H_{\text{Marr-Hildreth}}(\rho, \sigma) = (2/\sigma^2) \cdot [(\rho^2 / 2\sigma^2) - 1] \cdot \exp(-\rho^2 / 2\sigma^2)$$

(T : Seuil sur l'amplitude de la transition)

Si l'on se réfère aux applications de codage en relation avec les exigences définies au paragraphe 2.1.1, on voit qu'il est absolument essentiel de préserver la position de tous les contours existant dans la scène. Le filtre de Marr-Hildreth optimise le compromis détection-position mais il distord l'information des zones de l'image contenant beaucoup de contours proches (points anguleux, lignes contrastées). Ceci est dû essentiellement à l'opérateur de régularisation gaussien qui tend à avoir un effet passe-bas trop marqué sur l'image (cf. fig.2.11). Dans [39], Marr et Hildreth suggèrent de combiner les passages par zéro obtenus pour plusieurs valeurs d'écart type. Cependant aucun détail n'est donné sur la façon d'effectuer cette combinaison.

Dans le contexte des images traitées, on peut considérer qu'elles aient un rapport signal-sur-bruit élevé. Aussi, il semble plus opportun d'appliquer un simple opérateur d'estimation de dérivée seconde comme celui proposé par la relation (2.1). On a ainsi une plus grande précision bien que certains contours aient été perdus par rapport à la détection de Marr-Hildreth. Les figures 2.17 et 2.18 représentent les images du bâtiment et du caméraman filtrées avec $H(\rho, \theta)$. Les figures 2.19 et 2.20 donnent les images de contours correspondantes après poursuite comme décrit plus haut. Sauf indication contraire, ces images et celle de la figure 2.15d seront utilisées comme images de contrôle des contours dans la suite du travail.



Figure 2.17: Image du bâtiment filtrée par $H(p, \theta)$



Figure 2.18: Image du cameraman filtrée par $H(p, \theta)$.

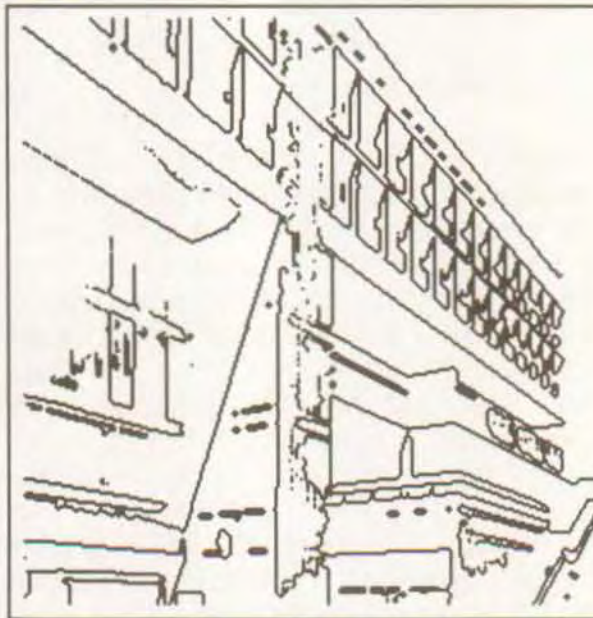


Figure 2.19: Contours détectés à partir de la fig. 2.17.

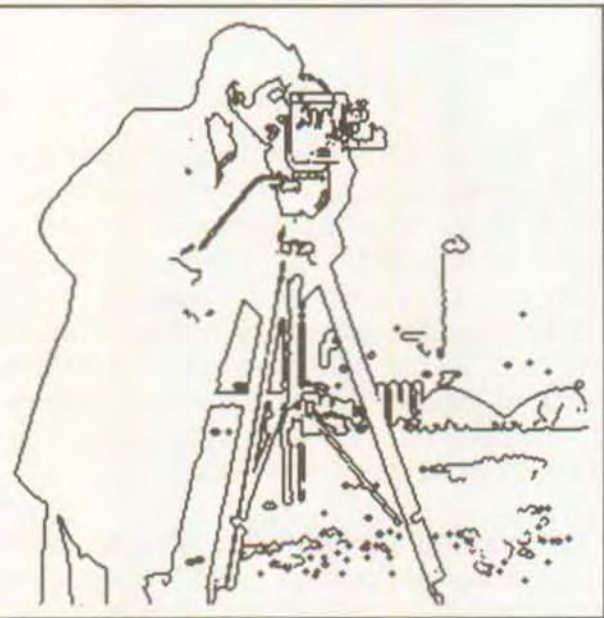


Figure 2.20: Contours détectés à partir de la fig 2.18.

Seuil de début de poursuite
Seuil de fin de poursuite

= 7.8% pour 2.19 et 19.6% pour 2.20.
= 6.6% pour 2.19 et 11.7% pour 2.20.

Les figures 2.21 et 2.22 donnent les images de contours que l'on obtient pour le bâtiment et le couple prétraités par le filtre médian 3×3 . Les résultats montrent une perte des détails comme il apparaît déjà à la figure 2.13. Des résultats similaires sont obtenus si l'on prétraite l'image avec le filtre de Nagao avant de la filtrer avec $H(p, \theta)$.



Figure 2.21: Contours détectés à partir de la fig. 2.13a.
Seuil de début de poursuite
Seuil de fin de poursuite

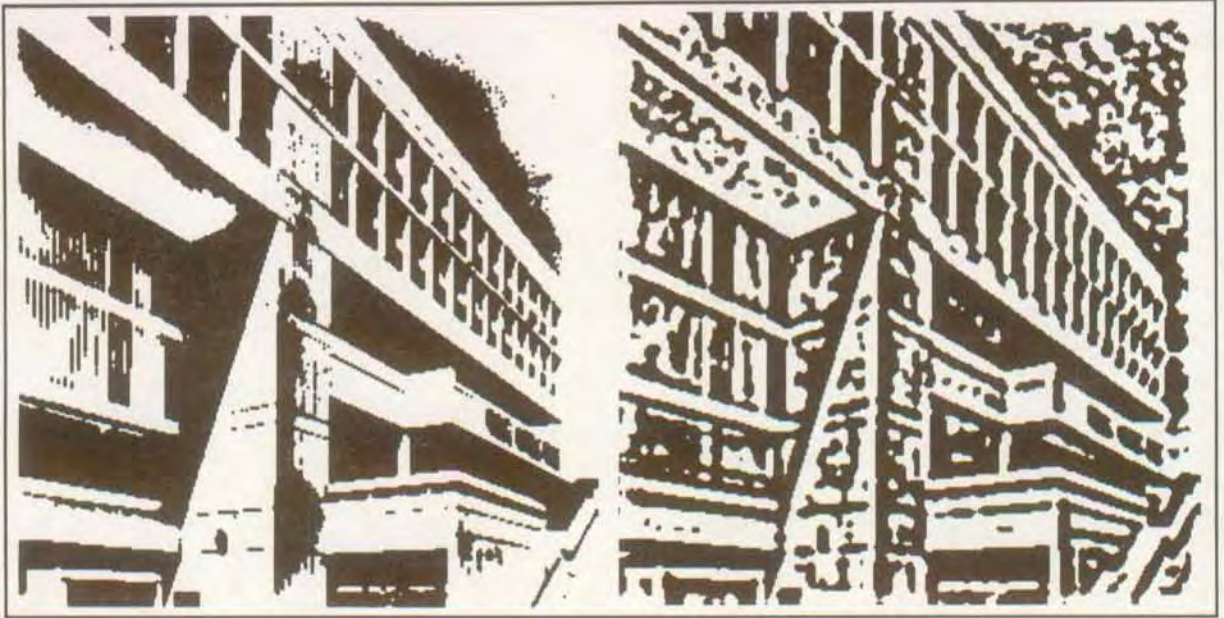
Figure 2.22: Contours détectés à partir de la fig 2.13c.
= 13.7% pour 2.21 et 19.6% pour 2.22.
= 5.9% pour 2.21 et 7.8% pour 2.22.

Si l'opération de détection de contour est déterminante pour préserver la sémantique de l'image lors de la segmentation, une autre image non moins importante peut être extraite pour regrouper certains pixels se trouvant d'un même côté d'un contour de l'image.

2.2.3 Image des structures positives et négatives :

Si l'on étudie la position des passages par zéro des images filtrées par des opérateurs de détection de contour basés sur une estimation de la dérivée seconde du signal, on remarque que les pixels situés de part et d'autre d'un tel contour et dans son voisinage, correspondent à des parties relativement homogènes du signal image. Comme il sera décrit au chapitre 4, cette partition peut se rendre utile pour faciliter le processus de rassemblement au voisinage des contours. Le nom de structures positives et négatives a été choisi en liaison avec le signe positif ou négatif de ces zones dans l'image filtrée.

Les figures 2.23a et 2.24a montrent le signe des images du bâtiment et du couple filtrées par $H(\rho, \theta)$. Les régions à forte activité, c'est-à-dire où le signe fluctue rapidement, ne relèvent pas d'information pertinente, les passages par zéro correspondants ne définissant pas de contours réels de l'image. Les figures 2.23b et 2.24b sont obtenues de la même façon pour les mêmes images de départ lorsqu'on a appliqué l'opérateur de Marr-Hildreth.

a) filtré par l'opérateur $H(\rho, \theta)$

b) filtré par l'opérateur de Marr-Hildreth

Figure 2.23: Image des structures positives et négatives du bâtimenta) filtré par l'opérateur $H(\rho, \theta)$

b) filtré par l'opérateur de Marr-Hildreth

Figure 2.24: Image des structures positives et négatives du couple

Une nouvelle fois, les images de contrôle obtenues à partir du filtre $H(\rho, \theta)$ seront utilisées pour piloter la segmentation, sauf autre indication.

Si l'on peut piloter la segmentation au moyen d'images de contrôle, il s'agit maintenant de définir les règles de modélisation du signal luminance pour représenter chaque zone homogène de l'image.

2.3 Approximation des données :

A la section 2.1, on a mis en évidence qu'il fallait une bonne correspondance entre l'approximation et le signal à l'intérieur de chaque région. Nous allons étudier ici le critère le plus approprié au calcul de cette approximation. Le choix d'une approximation au sens des moindres carrés va être justifié. L'analyse portera ensuite sur le genre de fonctions bidimensionnelles utilisées pour représenter le signal. Le point essentiel consiste à avoir une représentation aussi compacte que possible des données images pour chaque région, autrement dit un nombre très grand de points vont être décrits par une approximation avec peu de paramètres.

2.3.1 Type d'approximation :

La littérature abonde d'ouvrages sur la théorie des approximations [47-51]. Certains principes vont être repris ici mais il s'agit surtout de présenter les avantages de l'approximation au sens des moindres carrés dans le contexte de notre application. Les critiques généralement portées sur son utilisation pour des images naturelles proviennent du fait qu'elle a toujours été envisagée indépendamment d'une segmentation liée à la structure physique de la scène. Nous montrons que cette approximation permet une représentation efficace de la plupart des données d'une zone homogène de l'image. Les frontières de l'espace régulier sont détectées non pas par une mesure d'erreur quadratique moyenne mais par la perte ou non d'information contour. La figure 2.25 illustre ce principe.

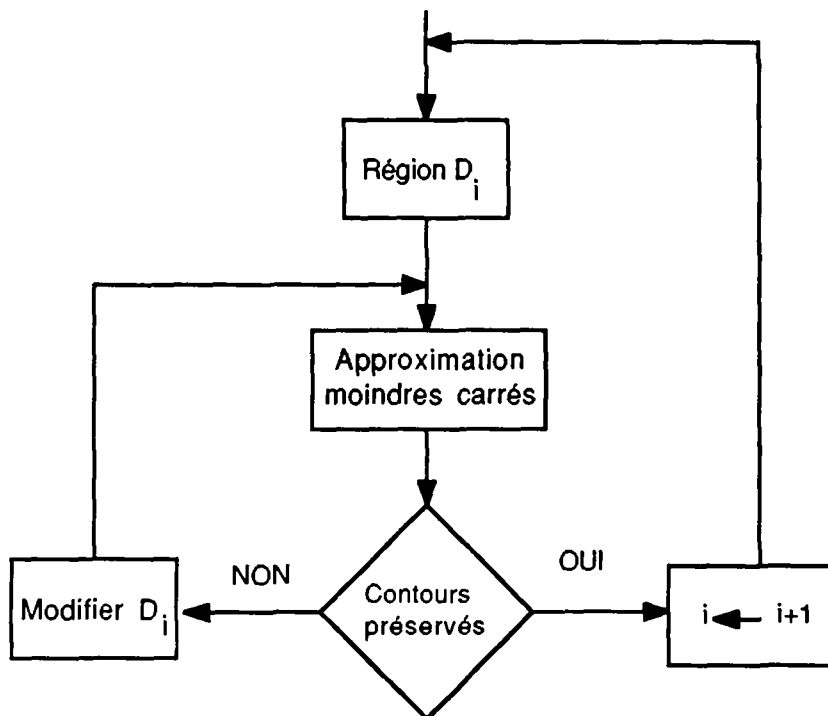
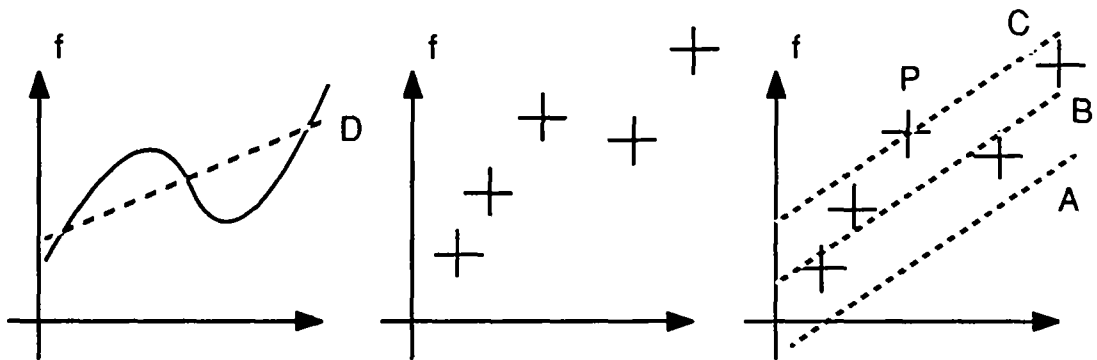


Figure 2.25: Principe de correspondance entre la segmentation et l'approximation.

Considérations générales:

Un exemple simple d'un problème d'approximation consiste à dessiner une droite qui suit au mieux une fonction continue f comme celle représentée à la figure 2.26a. Le même genre de problème peut être posé en remplaçant la fonction f par une série de points comme indiqué à la figure 2.26b. Trois choix possibles sont indiqués à la figure 2.26c. Subjectivement, on peut dire que les droites B et C sont meilleures que A. Entre B et C, tout dépend de l'importance relative que l'on veut donner au point P, d'où la nécessité d'une mesure de qualité.



a) Fonction f approximée par une droite D. b) Ensemble de points à approximer. c) Trois approximations par droites (A, B, ou C).

Figure 2.26: Exemple d'approximation d'une fonction f par une droite.

Dans cet exemple, les croix de la figure 2.26b indiquent les points à approximer. L'espace des fonctions d'approximation est défini par un ensemble de droites. Le critère d'approximation permet de choisir l'une de ces droites.

Trois considérations liées à tout problème d'approximation d'images ressortent de cet exemple :

(1) Quelle est la fonction $g(x,y)$ ou l'ensemble des données discrètes $g(k,l)$ à approximer, (k,l) correspondant à n'importe quel point du domaine bidimensionnel que l'on cherche à approximer ?

(2) Quel espace \mathbb{G} de fonctions g^{\wedge} d'approximation peut-on utiliser pour représenter ces données?

(3) Quel critère utiliser pour choisir une fonction g^{\wedge} appartenant à \mathbb{G} ?

Même si la réalité d'une image est un signal analogique, sa représentation numérique reste suffisante. Ainsi les données images peuvent être parfaitement considérées comme les données à approximer. L'utilisation d'une expression analytique $g(x,y)$ interpolant une série de pixels ne se justifie pas, cette fonction ne décrivant pas nécessairement l'évolution exacte de la luminance.

Si les données sont discrètes, les fonctions d'approximations peuvent être continues, les valeurs approximées ne rentrant en ligne de compte qu'aux lieux d'échantillonnage pour la mesure de la qualité de l'approximation. L'espace \mathbb{G} des

fonctions d'approximation est engendré par un ensemble de fonctions de base $\Psi_j(x,y)$ ($j=1, \dots, r$). Ainsi, on peut écrire :

$$g^{\wedge}(x,y) = \sum_{j=1}^r u_j \Psi_j(x,y) \quad (2.3)$$

ou sous forme vectorielle $g^{\wedge}(x,y) = \Psi^T(x,y) \cdot u$.

L'ensemble Ψ_j doit permettre d'engendrer toutes les caractéristiques possibles d'une zone homogène de l'image en jouant sur ses *paramètres* ou *coefficients* u_j .

Pour définir un critère d'approximation, il est nécessaire de définir une mesure de distance. Pour cela, considérons un espace vectoriel V muni d'une norme contenant g et g^{\wedge} , $d(a,b)$ définit la distance entre deux vecteurs a et b de cet espace si :

$$\begin{aligned} a \neq b &\Rightarrow d(a,b) > 0 \text{ et } d(a,b) = d(b,a) \\ a = b &\Rightarrow d(a,b) = 0 \end{aligned} \quad (2.4)$$

et si cette notion vérifie l'inégalité du triangle autrement dit :

$$\forall (a,b,c) \in V^3 \quad d(a,b) \leq d(a,c) + d(b,c) \quad (2.5)$$

Nous dirons que l'approximation g^{\wedge}_0 est meilleure que l'approximation g^{\wedge} du signal g si :

$$d(g^{\wedge}_0, g) < d(g^{\wedge}, g) \quad (2.6)$$

Pour tout point (k,l) du domaine D à approximer, on peut écrire :

$$g(k,l) = g^{\wedge}(k,l) + e(k,l) \quad (2.7)$$

$e(k,l)$ correspond à l'erreur commise en ce point. En général, $g(k,l)$ peut être formé d'une composante de bruit et d'une composante déterministe. Par $g^{\wedge}(k,l)$, on cherche à modéliser la composante déterministe, à caractériser ainsi l'espace régulier. Le choix des paramètres u_j va se faire de manière à minimiser une certaine fonction \mathbb{A} des erreurs d'approximation sur tous les points du domaine D . Ceci peut être formalisé en réécrivant l'équation (2.7) de la façon suivante :

$$e(k,l) = e(k,l, u_1, u_2, \dots, u_r) = g(k,l) - g^{\wedge}(k,l, u_1, u_2, \dots, u_r) \quad (2.8)$$

où l'on met en évidence que l'erreur est une fonction des paramètres d'approximation. Chercher le minimum de \mathbb{A} permet une transformation de l'espace des N pixels du domaine D dans l'espace des r paramètres u_j . En admettant que les pixels n'ont pas la même importance, cette fonction doit pondérer les erreurs

relatives à chacun des points. Il faut également déterminer la part respective des petites et grandes erreurs. Une expression assez générale de \mathbb{A} est donnée par :

$$\mathbb{A}(e(k,l); (k,l) \in D) = \sum_{(k,l) \in D} a(k,l) |e(k,l; u_1, u_2, \dots, u_r)|^p \quad (2.9)$$

où $a(k,l)$ et p sont des nombres réels positifs; D définit le domaine d'approximation.

Rechercher le minimum de cette fonction conduit à résoudre le système de r équations:

$$\partial \mathbb{A} / \partial u_j = 0 \quad j=1,2,\dots,r \quad (2.10)$$

$$p \sum_{(k,l) \in D} a(k,l) |e(k,l)|^{p-1} \cdot (\partial |e(k,l)| / \partial u_j) = 0$$

$$\sum_{(k,l) \in D} a(k,l) |e(k,l)|^{p-1} \cdot (\partial g^{\wedge}(k,l) / \partial u_j) \cdot (-\text{sgn}(e(k,l))) = 0$$

$$\text{où} \quad \text{sgn}(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x = 0 \\ -1 & \text{si } x < 0 \end{cases}$$

En adoptant une notation matricielle ($\mathbf{u}^T = [u_1 \ u_2 \ \dots \ u_r]$; $\mathbf{g}^T = [g(l_1) \ g(l_2) \ \dots \ g(l_N)]$; $\mathbf{g}^{\wedge T} = [g^{\wedge}(l_1) \ g^{\wedge}(l_2) \ \dots \ g^{\wedge}(l_N)]$; $\mathbf{e}^T = [e(l_1) \ e(l_2) \ \dots \ e(l_N)]$), on peut écrire les relations (2.9) et (2.10) de manière plus simple lorsque $p=2$ (approximation au sens des moindres carrés) :

$$\mathbf{A}(\mathbf{e}) = \mathbf{e}^T \mathbf{A} \mathbf{e} \quad (2.11)$$

où \mathbf{A} est la matrice diagonale $\text{diag}(a(k,l))$. l_j ($j=1,\dots,N$) définissent les N points de coordonnées (k,l) du domaine D .

La minimisation de la fonction \mathbb{A} revient à résoudre l'équation (voir Annexe A):

$$\partial \mathbb{A} / \partial \mathbf{u} = 0 \Leftrightarrow (\partial \mathbf{g}^{\wedge} / \partial \mathbf{u})^T \cdot \mathbf{A} \mathbf{e} = 0 \quad (2.12)$$

Si p est différent de 2, fixer les coefficients $a(k,l)$ et p de la fonction \mathbb{A} est une tâche difficile. Avant de le faire, nous allons décrire le critère d'approximation minimax ou L_{∞} qui consiste à minimiser l'erreur la plus grande sur l'ensemble des données à approximer.

Critère d'approximation minimax:

Ce critère veut minimiser la mesure de distance:

$$d_{\infty}(g, g^{\wedge}) = \max_{(k,l) \in D} |g(k,l) - g^{\wedge}(k,l)| \quad (2.13)$$

Cette mesure de distance peut être exprimée au moyen de l'expression (2.9) en faisant tendre p vers l'infini (voir infra). Pour minimiser d_{∞} , on peut procéder à la démarche itérative suivante:

a) choisir $r+1$ points $l_i \in D$ que l'on notera l_1, \dots, l_{r+1} ; on va s'arranger pour avoir la même valeur absolue e de l'erreur d'approximation en chacun de ces points,

b) résoudre le système de $r+1$ équations $g(l_i) = (-1)^i e + \sum_{(j=1..r)} u_j \Psi_j(l_i)$; l'erreur e et les coefficients u_j définissent les $r+1$ inconnues de ce système.

c) calculer, avec les paramètres u_j solution du système précédent, l'erreur aux points l_{r+2} à l_N restant du domaine D ,

d) chercher le maximum de ces erreurs E ,

e) si $E=e$, le processus s'arrête; sinon recommencer à partir de a) en remplaçant un des points l_1 à l_{r+1} par le point représentant l'erreur E .

On remarque que la solution optimum donne la même erreur e en chaque point du domaine à approximer. Ce critère est donc très sensible au bruit. Comme chaque région définissant une zone homogène contient nécessairement du bruit, ce genre d'approximation introduit une distorsion en tout point même où le bruit est presque inexistant. On n'arrive donc pas à reproduire les caractéristiques moyennes de la région. En outre, si l'on doit encore effectuer la partition de l'image, il devient très difficile de savoir comment modifier la région, l'erreur d'approximation e étant identique en chaque point. Ainsi, si l'on désire modifier un certain domaine, il faut pratiquement analyser parmi toutes les possibilités de modification celle qui produit la plus faible erreur d'approximation selon ce même critère L_{∞} . Enfin, la présence de contours à l'intérieur d'une région entraîne le plus souvent une approximation minimax présentant de faux contours, à cause de l'importante valeur de e de la solution optimale. En ce qui concerne le temps calcul, cette solution est très coûteuse (plusieurs inversions de matrices $(r+1) \times (r+1)$ selon le nombre d'itérations nécessaires), rendant sa mise en oeuvre pratique irréalisable.

Ce critère d'approximation s'avère donc inefficace pour modéliser les données images à l'intérieur d'un domaine D . Il s'agit alors d'étudier les effets des paramètres $a(k,l)$ et p de l'expression (2.9). Pour simplifier, appelons e_i les erreurs $e(l_i)$ en chaque point du domaine D .

Effet du paramètre p :

Si sa valeur passe de p_1 à p_2 avec p_2 plus grand que p_1 , la minimisation de \mathbb{A} entraîne une diminution des plus grandes valeurs e_i obtenues avec une solution optimale utilisant p_1 . Ceci correspond d'un autre côté à une augmentation des autres erreurs, de telle sorte que la dynamique des erreurs possibles tend à se réduire au fur et à mesure que p devient grand. A la limite, lorsque p tend vers

l'infini, on retrouve la solution minimax qui donne une erreur identique en chaque point. De la même manière, l'approximation optimale pour de faibles valeurs de p tend à garder la plupart des erreurs faibles. Si une région ne contient pas de contours, on est ainsi assuré d'avoir une bonne correspondance entre le signal original et son approximation à l'intérieur du domaine analysé. Par contre, en présence de contours, il n'y a correspondance que si l'on a à faire à de petites régions sans quoi, il faut modifier la forme de la région considérée jusqu'à faire épouser l'une de ses frontières avec le contour en question. Si de faibles valeurs de p semblent ainsi bien adaptées, la résolution du système (2.10) est fortement simplifiée en choisissant p égal à 2. En effet, il suffit ainsi de résoudre le système d'équations (2.12).

Effet des coefficients de pondération $a(k,l)$:

Vu que l'information contour est de première importance, il semble nécessaire de forcer l'approximation à suivre le signal original au voisinage des contours. Bien entendu, il s'agit plus de préserver la position du saut de luminance que son amplitude. C'est pourquoi, si une certaine région contient un contour il s'agira de pondérer fortement les erreurs en fonction du gradient du signal original. Néanmoins, ceci ne doit pas se faire au détriment d'une certaine fidélité dans la reproduction du signal à variation lente de part et d'autre du contour. On risquerait en outre de créer des contours artificiels, violant ainsi les exigences de départ. Ces artefacts sont fortement fonction de l'espace des configurations qui peut être engendré par la modification des coefficients d'approximation u_j . En utilisant des fonctions d'approximations polynomiales, on a pu relever que dans le contexte d'une grande région contenant un contour:

1. toute pondération fonction du gradient tend à détruire l'information de part et d'autre du contour pour des polynômes de degré inférieur ou égal à 4 (conjecture).

2. une représentation du saut de luminance est impossible si aucune pondération n'est effectuée. Dans un tel cas de non pondération, on remarque une erreur importante au voisinage du contour (conjecture).

La première conjecture justifie donc le choix d'éviter de pondérer les erreurs d'approximation selon leur position dans la région considérée. Ceci présente d'autant plus l'avantage de simplifier les calculs. Le contenu de la deuxième conjecture ne peut être évité qu'en modifiant la forme de la région de manière à faire épouser certaines de ces frontières avec le contour. Les mesures d'erreur le long de ces contours permettent de changer la partition de l'image comme on le verra à la section suivante.

Approximation au sens des moindres carrés:

Le critère d'approximation retenu correspond donc à minimiser la somme des erreurs quadratiques. La solution optimale peut être obtenue en reprenant les relations (2.10) et (2.12), par la résolution du système

$$\mathbf{z}^T (\mathbf{g} - \mathbf{Z} \mathbf{u}) = 0 \quad (2.14)$$

où \mathbf{Z}^T est la matrice $r \times N$ $[\Psi(l_1) \Psi(l_2) \dots \Psi(l_N)]$.

En effet, le vecteur des valeurs approximées peut s'exprimer par $\mathbf{g}^{\wedge} = \mathbf{Z}\mathbf{u}$. La relation (2.14) donne comme vecteur optimal de coefficients:

$$\mathbf{u} = (\mathbf{Z}^T\mathbf{Z})^{-1} \cdot \mathbf{Z}^T \mathbf{g} \quad (2.15)$$

Pour simplifier, nous noterons \mathbf{S} la matrice symétrique définie positive $r \times r$ $\mathbf{Z}^T\mathbf{Z}$ et \mathbf{h} le vecteur $\mathbf{Z}^T\mathbf{g}$. L'équation (2.15) ne peut être résolue que si \mathbf{S} est non singulière. L'élément général de cette matrice s'écrit:

$$s_{pq} = \sum_{(k,l) \in D} \Psi_p(k,l) \cdot \Psi_q(k,l) \quad (2.16)$$

Lorsque une solution unique au problème existe, la résolution de (2.15) requiert:

- 1.- l'évaluation des vecteurs $\Psi(k,l)$ pour tous les points du domaine D ,
- 2.- $Nr(r+1)/2$ multiplications et $(N-1)r(r+1)/2$ additions pour calculer la matrice \mathbf{S} . Pour simplifier, on considèrera que $O(Nr^2)$ opérations suffisent,
- 3.- La multiplication matrice-vecteur pour estimer le vecteur \mathbf{h} ce qui implique $O(Nr)$ opérations,
- 4.- la résolution du système $\mathbf{S} \cdot \mathbf{u} = \mathbf{h}$, ce qui correspond en général à $O(r^3)$ opérations [52].

On voit donc que les étapes 1 à 3 sont les plus coûteuses comme en général $N \gg r$. Pour pouvoir réaliser la modification successive de la segmentation en évaluant chaque fois la meilleure approximation au sens des moindres carrés, on étudiera aux chapitres suivants comment il est possible d'optimiser ce traitement, en tenant compte de diverses propriétés de ce genre d'approximation.

2.3.2 Fonctions d'approximation:

Le choix des fonctions Ψ_j est fondamental. En effet, c'est elles qui permettent de générer l'espace des approximations possibles en jouant sur les coefficients u_j .

Le choix d'un ensemble complet de fonctions orthogonales ne se justifie pas car la méthode proposée implique la modification systématique de la forme des régions. Ceci entraîne une évolution de l'ensemble des fonctions orthogonales sur chaque région. Si le calcul de l'approximation est simplifié, la matrice \mathbf{S} pouvant se ramener à l'identité après normalisation, une procédure d'orthogonalisation [53] de l'ensemble des fonctions devient nécessaire. En outre, aucune considération a priori ne permet de choisir un sous-ensemble r de ces fonctions pour calculer l'approximation. Il faudrait estimer la transformation de l'espace image dans l'espace des N fonctions orthogonales sur le domaine D et garder les r coefficients

les plus importants. Au niveau d'un système de codage, il devient alors nécessaire de transmettre pour chaque région les r fonctions correspondantes retenues.

A partir d'un ensemble de r fonctions de base Ψ_j , rien ne sert de les orthogonaliser sur le domaine D où l'on désire calculer l'approximation. En effet, cette transformation correspond à un simple changement de base, si bien que la fonction approximée g^{\wedge} possède exactement la même expression sur D . Ceci est démontré dans l'annexe B. Les résultats de la méthode de segmentation sont donc identiques à ceux obtenus sans orthogonalisation de l'ensemble des fonctions de base, si ce n'est que les calculs sont compliqués davantage.

Comme on suppose que dans la segmentation finale, il n'existe pas de régions contenant des contours, il est raisonnable de dire que le signal à l'intérieur de ces régions est à variation lente, selon les hypothèses formulées au paragraphe 2.2.1.

Les fonctions polynomiales sont bien adaptées à représenter de telles variations. En outre, la régularité des surfaces engendrées par des polynômes est grande: cela évite d'avoir de phénomènes d'oscillations désagréables sur le plan visuel, que l'on aurait en utilisant des fonctions orthogonales périodiques (Fourier, Hadamard,...). Si une bonne correspondance est assurée sur une grande partie du domaine d'approximation, les fonctions polynomiales présentent l'avantage de s'éloigner rapidement aux limites de ce domaine, garantissant la non destruction de contours existants.

Nous admettrons donc de représenter les différentes régions au moyen de polynômes bidimensionnels. On peut écrire généralement:

$$\Psi_j(x,y) = x^p \cdot y^q \quad (2.17)$$

Le nombre de ces fonctions reste faible de manière à "transformer" beaucoup de pixels en peu de coefficients d'approximation. On garantit ainsi une bonne simplification de la scène ainsi qu'une compression plus élevée pour un nombre donné de régions. En maintenant r relativement bas, on réduit également la complexité de calcul vu que celle-ci croît avec r . Les monômes de base exprimés par la relation (2.17) ne doivent pas être de degré trop élevé pour éviter des erreurs de troncation dans le calcul de l'approximation. Pour pouvoir décrire une grande variété de surfaces à variation lente, on utilisera des fonctions polynomiales de degrés 0 à 3 impliquant respectivement 1,3,6 et 10 coefficients. Plus précisément, on peut écrire :

$$\begin{aligned} P_0(x,y) &= u_1 \\ P_1(x,y) &= u_1 + u_2x + u_3y \\ P_2(x,y) &= u_1 + u_2x + u_3y + u_4x^2 + u_5xy + u_6y^2 \\ P_3(x,y) &= u_1 + u_2x + u_3y + u_4x^2 + u_5xy + u_6y^2 + u_7x^3 + u_8x^2y + u_9xy^2 + u_{10}y^3 \end{aligned} \quad (2.18)$$

Nous verrons au chapitre suivant comment il peut être intéressant d'avoir des fonctions polynomiales entraînant une formulation séparable du problème.

2.3.3 Quantification de l'approximation:

Une fois les coefficients d'approximations optimaux calculés, on estime l'approximation g^{\wedge} en chaque point de la région considérée. Si la dynamique du signal original à 8 bits est fixée entre 0 et 255, il s'agit d'y inclure également celle de l'approximation g^{\wedge} . Si la dynamique de g^{\wedge} dépasse celle du signal original, une première possibilité consiste à requantifier linéairement le signal approximé en 256 niveaux. Néanmoins, on introduit ainsi une erreur en chaque point. Il semble donc plus judicieux de tronquer les points se trouvant au-delà des seuils de luminance de manière à les ramener à ces valeurs. On arrondit les valeurs des autres points aux entiers les plus proches. Si la dynamique du signal approximé s'étend de α à β avec $\alpha \leq 0$ et $\beta \geq 255$,

- par quantification linéaire entre 0 et 255, on commet en chaque point une erreur:

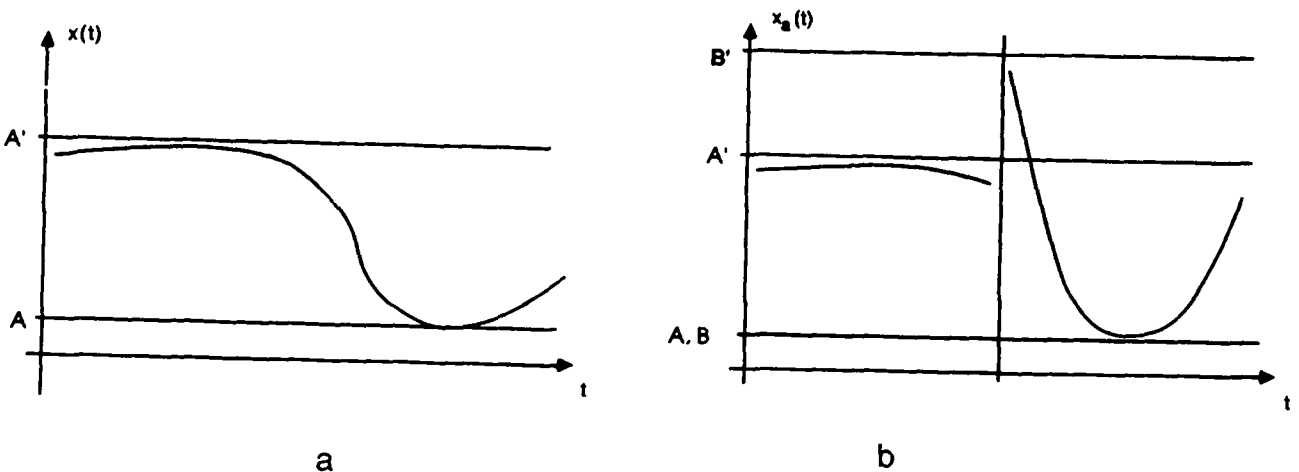
$$e_q = |g - 255.(g^{\wedge} + \alpha) / (\beta - \alpha)| \quad (2.19)$$

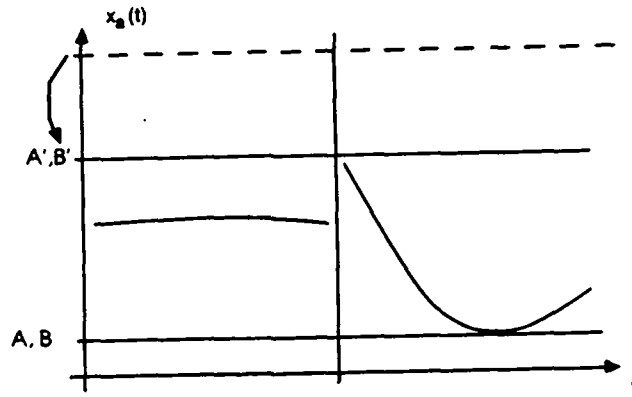
au lieu de $|g - g^{\wedge}|$. On risque au plus d'augmenter l'erreur de

$$|255.(g^{\wedge} + \alpha) / (\beta - \alpha) - g^{\wedge}| + 0.5 = |g^{\wedge}(1 - 255 / (\beta - \alpha)) + 255\alpha / (\beta - \alpha)| + 0.5 \quad (2.20)$$

Le terme 0.5 provient de l'arrondissement de la valeur absolue à l'entier le plus proche.

- par troncation des valeurs sortant de l'intervalle $[0, 255]$, on augmente l'erreur en chaque point compris dans cette gamme au plus de 0.5 alors qu'elle est réduite de α pour les valeurs en-dessous de 0 et de $\beta - 255$ pour celles au-dessus de 255. Il y a diminution de ces valeurs car le signal de départ était compris entre 0 et 255. Cette deuxième solution sera donc retenue. La figure 2.27 illustre par ailleurs les problèmes de perte de contraste qui peuvent résulter du choix de la quantification linéaire.





c

Figure 2.27: Illustration de la perte de contraste résultant d'une requantification linéaire.

En 2.27a, on présente un signal original et sa gamme dynamique AA'. En 2.27b, on a une représentation de l'approximation du signal original sur deux régions au moyen de deux arcs de parabole; la dynamique correspondante s'étend sur le segment BB'. Enfin, en 2.27c apparaît la perte de contraste résultant de la requantification linéaire de BB' à AA'.

Le problème de l'approximation ayant été résolu, il s'agit maintenant de définir les critères nécessaires à estimer la qualité de la segmentation pour en justifier une éventuelle modification.

2.4 Mesures de qualité :

Deux indices de qualité seront mis en évidence dans ce qui suit. Le premier est basé sur des estimations globales telles erreur quadratique alors que le second définit une mesure plus contextuelle basée sur la bonne représentation de l'information contour. Une stratégie évolutive doit être mise sur pied pour tenir compte de la structure des régions dans le choix des mesures d'erreur.

2.4.1 Mesures globales:

Considérons pour ce faire l'erreur quadratique (EQ) sur le domaine D analysé. Celle-ci s'exprime par:

$$\begin{aligned} EQ &= (\mathbf{g} - \mathbf{Z} \mathbf{u})^T (\mathbf{g} - \mathbf{Z} \mathbf{u}) \\ &= \mathbf{g}^T \mathbf{g} - 2 \mathbf{u}^T \mathbf{Z}^T \mathbf{g} + \mathbf{u}^T \mathbf{Z}^T \mathbf{Z} \mathbf{u} \end{aligned} \quad (2.21)$$

où \mathbf{u} représente le solution optimale de l'approximation au sens des moindres carrés sur D. En utilisant (2.15), on voit que $\mathbf{Z}^T \mathbf{Z} \mathbf{u} = \mathbf{S} \mathbf{u} = \mathbf{Z}^T \mathbf{g} = \mathbf{h}$. La relation (2.19) se ramène dès lors à :

$$EQ = \mathbf{g}^T \mathbf{g} - \mathbf{u}^T \mathbf{h} \quad (2.22)$$

Cette mesure de qualité ne donne qu'une estimation globale de la représentativité de l'approximation sur D . De manière à ne pas perdre l'information contour, il faut rester sévère quant au seuil à ne pas dépasser. Comme le montreront les résultats de découpage présentés au chapitre suivant, ceci entraînera une segmentation de l'image en un nombre élevé de petites régions.

On peut diminuer ce nombre en rendant ce critère de qualité relativement indépendant de la taille de la région. Il suffit de pondérer l'erreur quadratique par les N points du domaine aboutissant ainsi à ce qui est appelé l'erreur quadratique moyenne (EQM). On a donc la relation $EQM = EQ/N$. Par rapport à EQ , la segmentation donne une partition de l'image avec des régions de plus grande taille, mais il faut toujours placer le seuil relativement bas de manière à préserver les contours. Il semble donc nécessaire d'introduire des mesures de qualité plus contextuelles qui tiendront compte de la présence de contours dans le domaine analysé. Par opposition aux critères considérés jusqu'ici, ces mesures seront évaluées localement.

2.4.2. Mesures contextuelles:

Pour mesurer la qualité de reproduction des contours, il faut dans un premier temps utiliser l'image de contrôle des contours pour déterminer l'emplacement de ceux-ci. Ensuite, une erreur de contour (EC) est estimée. On évalue pour cela l'erreur commise au niveau des pixels se trouvant de part et d'autre du contour. De manière à rendre cette mesure indépendante de la longueur des contours apparaissant dans l'image on pondère les erreurs par le nombre de points impliqués par cette mesure. On peut ainsi écrire:

$$EC = (1/N_C) \cdot \sum_{(k,l) \in (D \cap C)} e^2(k,l) = (1/N_C) \sum_{(k,l) \in (D \cap C)} (g(k,l) - g^{\wedge}(k,l))^2 \quad (2.23)$$

où C représente l'ensemble des points situés de part et d'autre de tous les contours de l'image. N_C définit le nombre de points de l'intersection $(D \cap C)$. Bien entendu, l'efficacité de cette mesure dépend de l'exactitude de l'image de contrôle des contours.

Une autre possibilité consiste à pondérer les erreurs au voisinage des points contours par une estimation du gradient de l'image en ces points. Dans ce cas, la dégradation peut être calculée en partant de la formule (2.10) avec $a(k,l)$ correspondant à la composante du gradient normale au contour.

Si EQ ou EQM permettent de voir la correspondance moyenne qui existe entre le signal original et son approximation, EC détermine si des contours réels sont perdus. Ces critères sont élaborés pour pouvoir mesurer la qualité de la segmentation et non celle de l'approximation.

Définir un seuil de qualité acceptable est une tâche difficile si l'on n'a pas d'idée a priori de ce qu'on est en train d'analyser. La technique adaptative de rassemblement permet d'éviter ceci car on regroupe à chaque fois les deux régions présentant le moins de dissimilarité. Il reste à définir cette dissimilarité.

Si la présence de contours justifie la définition de mesures appropriées, il en va de même des mesures en liaison avec la forme et la taille des régions. En effet, selon les régions impliquées dans le processus de rassemblement, il apparaît nécessaire de faire évoluer les critères permettant d'estimer la ressemblance entre ces régions. La section 4.5. définira la stratégie à adopter en liaison avec la structure des régions. Il faudra auparavant analyser l'effet des différentes mesures de dissimilarité envisagées.

Commentaires:

1) Le premier terme de (2.22) correspond simplement à la somme des carrés des valeurs de l'image sur le domaine D . C'est donc l'énergie de ce signal sur ce domaine. Le second $\mathbf{u}^T \mathbf{h} = \mathbf{g}^T \mathbf{g}$ définit quant à lui l'énergie du signal d'approximation. Ainsi, si l'on adopte le critère des moindres carrés, l'erreur quadratique est exprimée par la différence d'énergie du signal original et de son approximation.

2) Le calcul de l'énergie du signal d'approximation nécessite uniquement $O(r)$ opérations, les vecteurs \mathbf{u} et \mathbf{h} ayant été estimés lors du calcul de l'approximation. On évite ainsi le calcul point par point du signal approximé sur D , qui entraînerait $O(Nr)$ opérations supplémentaires.

3) Le calcul de (2.23) requiert l'évaluation de N_c pixels approximés, ce qui peut être réalisé en $O(N_c r)$ opérations.

2.5 Conclusion :

Ce chapitre a souligné l'importance d'une approche adaptative pour segmenter une image. En effet, la nécessité de préserver l'information contour en recherchant les espaces réguliers de l'image ne peut se faire qu'en adaptant itérativement la forme des régions et de l'approximation en liaison avec les données originales. De manière à limiter la complexité d'analyse, on a proposé une stratégie de division et rassemblement dont les détails vont être présentés dans ce qui suit.

Un prétraitement a été mis en oeuvre de manière à extraire des images de contrôle des contours servant à piloter les différentes phases de la segmentation. Le choix du critère des moindres carrés pour approximer les données a été justifié dans le contexte de la segmentation d'images. Le signal luminance de chaque région est reconstruit en utilisant des polynômes bidimensionnels en raison de la correspondance qui existe entre des portions de scènes naturelles et ce genre de fonctions. Enfin, ce chapitre a servi à mettre en évidence l'utilisation de mesures globales et contextuelles pour estimer la qualité de la segmentation. Les premières donnent la correspondance moyenne qui existe entre le signal original et son approximation à l'intérieur d'une région. Les secondes garantissent de préserver les contours de l'image et l'élaboration correcte de la procédure de rassemblement en liaison avec la structure des régions.

3. Opération de division

Ce chapitre présente la première phase de la segmentation : la division ou le découpage. L'image de départ est *subdivisée de manière adaptative en une série de carrés* de différentes tailles qui seront ensuite regroupés lors du processus de rassemblement.

Chaque carré définit une partie d'un espace régulier de l'image. Il s'agit donc de garantir tout au long du processus de division une bonne correspondance entre les données originales et l'approximation à l'intérieur de chacun de ces carrés. Pour ce faire, on estimera la qualité de l'approximation au moyen de l'erreur quadratique moyenne (mesure globale) et de l'erreur le long des contours présents dans le carré (mesure contextuelle).

Le chapitre est organisé en six sections. Dans un premier temps, on définit les critères de subdivision de l'image originale. On s'attache ensuite à décrire comment structurer les données pour implanter l'algorithme de division. La troisième section présente la complexité de calcul du problème et montre comment il est possible d'optimiser les différentes phases du traitement. La quatrième section décrit comment représenter l'information après division. Dans la cinquième section, on analyse expérimentalement les effets des différents paramètres de l'algorithme en appliquant celui-ci à plusieurs images naturelles. Cette partie du chapitre donne un aperçu de la compression de l'information qui est déjà réalisée à ce niveau du traitement. La dernière section résume l'exposé qui précède.

3.1 Principes généraux :

Comme il a été décrit au chapitre précédent, chaque région résultant de la segmentation de l'image est représentée par une approximation au sens des moindres carrés au moyen d'un ensemble de fonctions polynomiales bidimensionnelles.

Si l'on part d'une image originale de taille $2^q \times 2^q$, il s'agit de mesurer la qualité de l'approximation à l'intérieur de celle-ci. Si celle-ci est insuffisante, on subdivise l'image en quatre carrés de taille $2^{q-1} \times 2^{q-1}$. On calcule ensuite pour chacun de ces carrés l'approximation au sens des moindres carrés au moyen du même ensemble de monômes $\Psi_j(x,y)$. La qualité de cette approximation est alors estimée pour chacun d'eux. Si celle-ci est acceptable, le carré correspondant apparaîtra comme définitif dans la division. Ce carré définira un espace ou une partie d'espace régulier de l'image. Si la qualité est par contre jugée insuffisante, on recommence le processus de subdivision pour le carré correspondant. Ce principe est illustré à la figure 3.1. Le première taille d'analyse ($2^q \times 2^q$) définit ce que l'on appelle le premier *niveau* de traitement. De la même façon, les tailles inférieures ($2^{q-1} \times 2^{q-1}$, $2^{q-2} \times 2^{q-2}$, ...) définissent les niveaux de traitement suivants (2, 3, ...).

La division doit nécessairement se terminer puisqu'on ne peut aller au-delà de la résolution de l'image (carré de taille 1×1 = dimension d'un pixel). En réalité, il s'arrête avant, car le polynôme bidimensionnel va interpoler les pixels originaux du carré dès que la taille de celui-ci va devenir inférieure ou égale au nombre r de coefficients du polynôme.

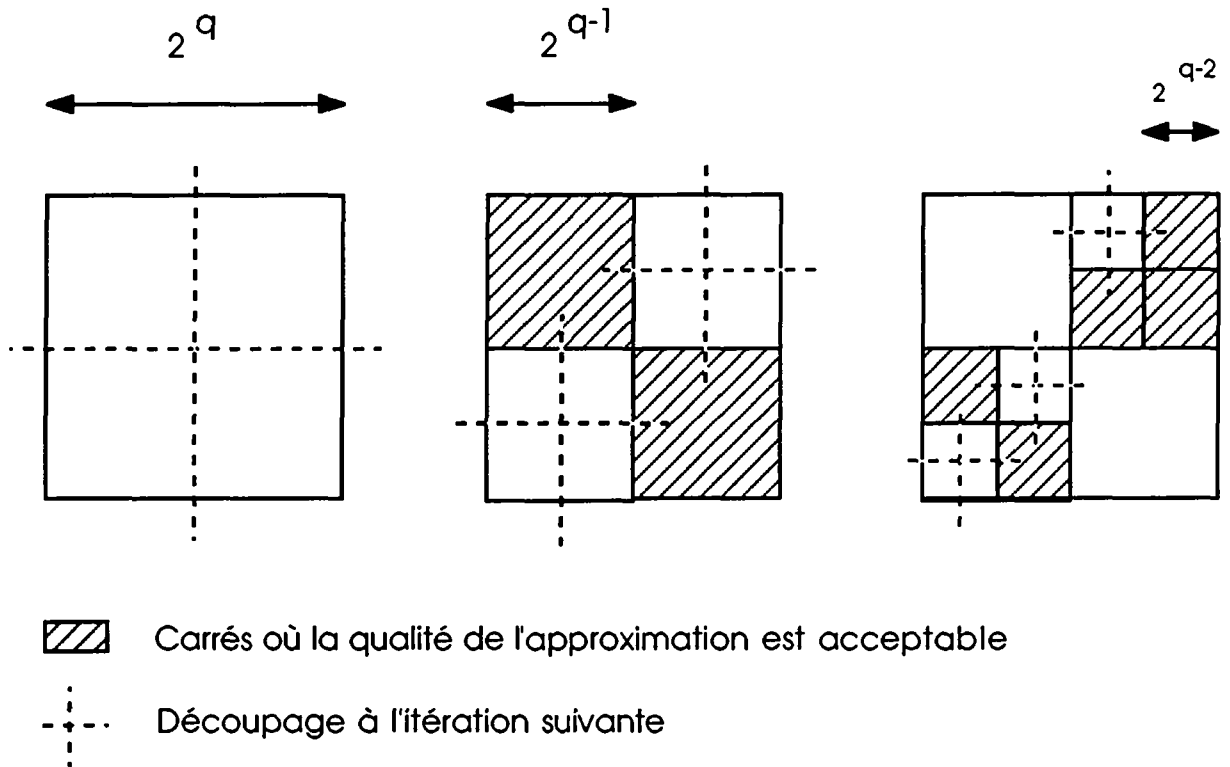


Figure 3.1: Principe de division.

Après la division, la partition de l'image est définie par une série de carrés ayant sur chaque côté un nombre de pixels constituant une puissance de 2.

La qualité de l'approximation est estimée sur la base des critères EQM et EC (voir section 2.4). Pour tenir compte de l'adéquation moyenne sur un carré donné on utilise l'erreur quadratique moyenne (EQM). Pour préserver les contours de l'image dans le carré considéré, on fait appel à l'erreur de contour (EC). Sur chacune de ces mesures seront placés des seuils d'acceptabilité, respectivement T_1 et T_2 . Dès que l'un des deux est franchi, la division sera effectuée, comme décrit précédemment. Le seuil placé à T_2 sera faible alors qu'il restera modérément élevé pour T_1 , ceci en liaison avec les hypothèses de non-dégradation de l'information contour et de dégradation tolérée pour la reproduction de l'information texture (voir paragraphe 2.1.1). La figure 3.2 reproduit l'algorithme de division dans son ensemble.

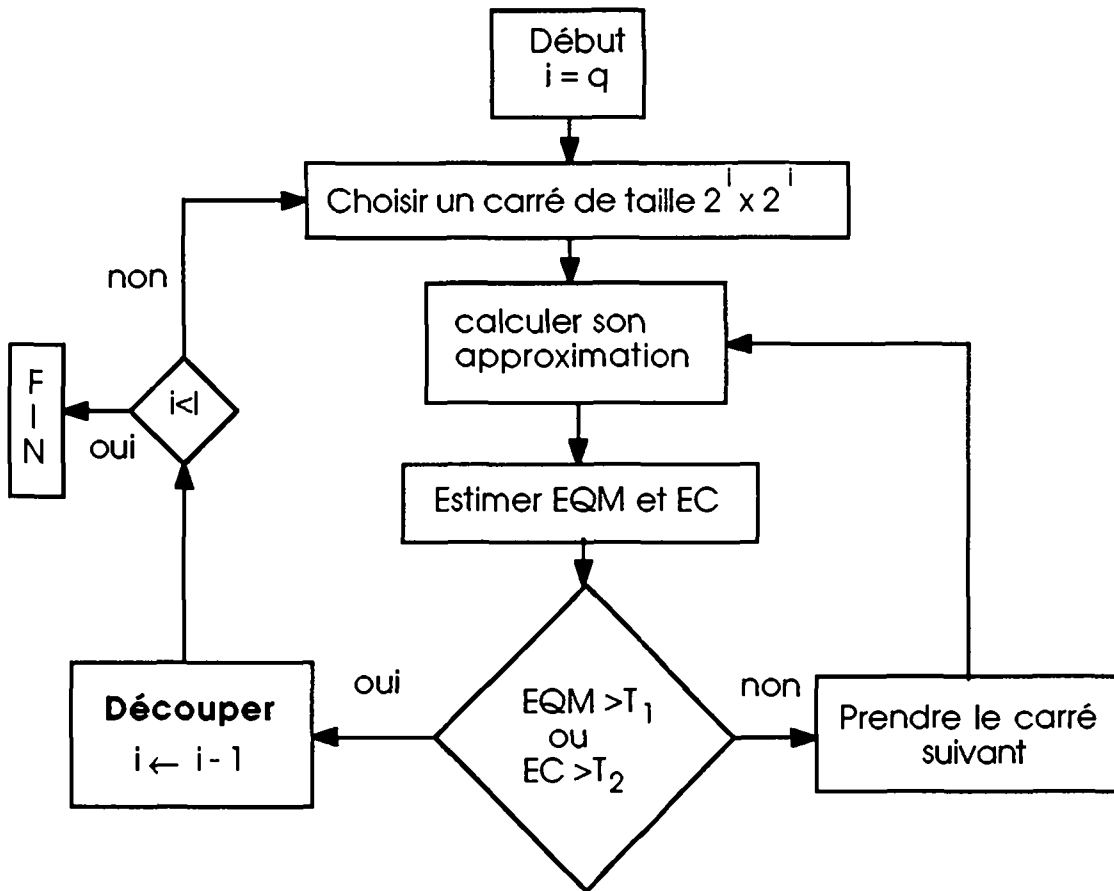


Figure 3.2: Algorithme de division (l : plus petite puissance de 2 admise selon (3.1)).

Dans les paragraphes qui suivront, différents commentaires sont apportés au principe de l'algorithme de division qui vient d'être décrit.

3.1.1 Existence d'une solution dans le calcul de l'approximation:

De manière à éviter de rendre le calcul de l'approximation impossible, la matrice \mathbf{S} devenant singulière, on a décidé de limiter le processus de subdivision à la taille de carré immédiatement supérieure à une indétermination du système (2.15). Le plus petit carré aura au plus $2^l \times 2^l$ points, avec :

$$2^l \times 2^l \leq r \quad (3.1)$$

La matrice \mathbf{S} ne présentera jamais de singularité pour des régions carrées de tailles plus grandes que $2^l \times 2^l$ si les polynômes usuels (de degré 0, 1, 2, ...) sont utilisés pour approximer les données de départ. Ceci est démontré dans l'annexe C.

3.1.2 Formes des régions après division:

Il est à priori envisageable de subdiviser l'image originale en d'autres formes géométriques que des carrés. Prendre des carrés s'explique par simplicité d'une part, par le fait que les images de départ sont en général de taille carrée et que la

grille d'échantillonnage est carrée. Disons encore que la prédominance de contours horizontaux et verticaux dans les images naturelles privilégie le choix de décomposer l'image en un ensemble de régions présentant des frontières de mêmes directions. Le principe de l'algorithme de division reste inchangé si l'on décidait de diviser l'image en des régions de même forme non carrées, par exemple des triangles isocèles rectangles (Des problèmes peuvent seulement surgir si l'on désire descendre au niveau d'un pixel pour la plus petite taille de région, celle-ci étant nécessairement carrée).

Le fait de subdiviser chaque carré en quatre sous-carrés de taille deux fois plus petite permet de passer *progressivement* d'une analyse globale de l'image à son analyse locale. Une subdivision en un nombre plus élevé de régions serait défavorable s'il existe des parties d'espaces réguliers de plus grande taille. Une division en deux rectangles identiques compliquerait l'algorithme car il faudrait tester les deux décompositions possibles de manière à choisir celle qui conduit à la meilleure reproduction des données de départ, ceci pour autant qu'un découpage ultérieur ne s'avère pas nécessaire.

Il faut que dans la partition finale, l'image soit décomposée en un ensemble de régions de n'importe quelle forme et taille, il semble nécessaire de subdiviser l'image jusqu'au niveau d'un pixel. Cette subdivision ultérieure sera réalisée uniquement pour les carrés contenant des contours en utilisant l'image de contrôle des structures positives et négatives définie au paragraphe 2.2.3. Dans le processus de rassemblement, on garantira la précision requise en traitant séparément l'information relative à ces carrés contenant des contours.

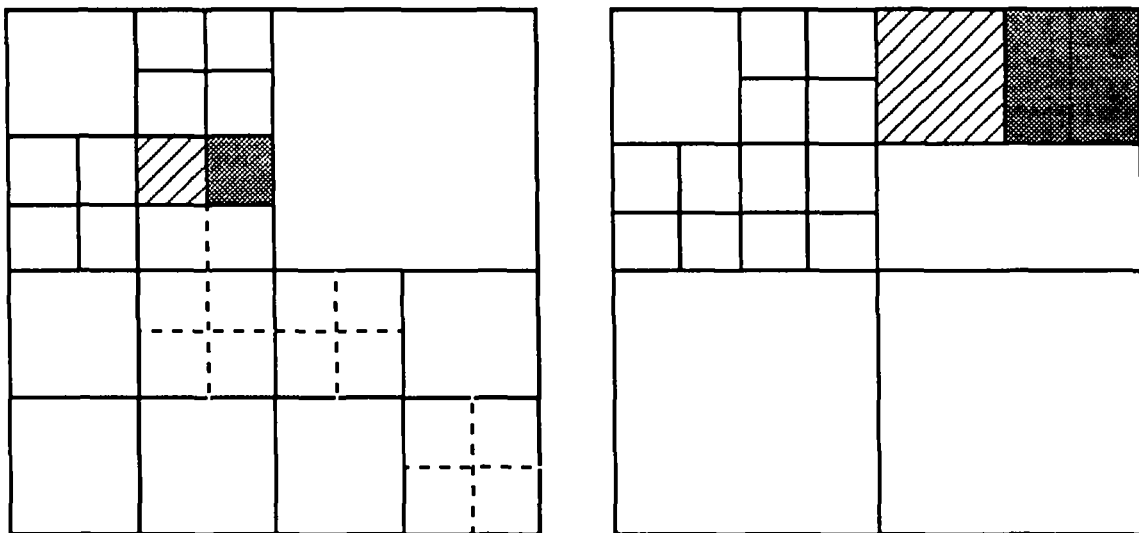
Après avoir décrit les principes de la division, la section suivante présente la structure des données nécessaire à une implantation de l'algorithme.

3.2 Implantation de l'algorithme:

Deux solutions sont proposées. La première solution permet de traiter tous les carrés d'un même niveau simultanément. La seconde fait appel à la notion de récursivité. L'analyse de chaque carré est alors effectuée cycliquement du premier niveau (image) au niveau le plus élevé (plus petite taille de carré) selon un balayage prédéfini.

Dans le premier cas, on traite tous les carrés d'une même taille simultanément alors que dans le deuxième cas, on cherche systématiquement à diviser un carré avant de considérer un autre carré.

La figure 3.3a illustre un état intermédiaire de l'algorithme lorsqu'on utilise la première possibilité; La figure 3.3b fait de même si l'on choisit la méthode récursive.



a) Analyse hiérarchique (les carrés à analyser de même taille que le carré courant sont indiqués en traitillé).

b) Analyse descendante-ascendante cyclique.

Figure 3.3: Etat intermédiaire de l'algorithme de division.
Le carré couramment analysé est indiqué en hachuré.
Celui qui sera analysé à l'étape suivante est indiqué en gris.

Ces deux approches donnent les mêmes résultats; elles ne définissent que deux implantations différentes de l'algorithme.

3.2.1 Analyse hiérarchique:

On commence l'analyse au premier niveau, c'est-à-dire $2^q \times 2^q$. Si une division est nécessaire, on mémorise dans une liste les coordonnées respectives du premier point en haut à gauche de chaque carré résultant de la subdivision, en l'occurrence $(1,1)$, $(1,1+2^{q-1})$, $(1+2^{q-1},1)$ et $(1+2^{q-1},1+2^{q-1})$. Ces coordonnées vont caractériser le carré. On répète alors la procédure pour tous les carrés restant à analyser. Si une division s'avère nécessaire, on insère *en fin de liste*, les coordonnées associées aux quatre carrés de taille immédiatement inférieure résultant de la subdivision du carré analysé. A chaque fois qu'un carré est analysé, on l'élimine de la liste. L'algorithme se termine dès que la liste est vide. De manière à savoir quelle est la dimension d'un carré dans la liste, il suffit d'utiliser deux compteurs. Le premier indique le nombre de carrés de taille donnée restant à analyser dans la liste, le second, le nombre de carrés de taille moitié en fin de liste. Chaque fois qu'un carré est analysé le premier compteur est décrémenté d'une unité; s'il y a division, le second est incrémenté de quatre unités. Dès que le premier compteur arrive à zéro, on échange les valeurs des deux compteurs. On sait qu'à partir de ce moment on travaille avec des carrés de taille moitié de la taille qu'on a eu jusqu'à ce que le premier compteur arrive à zéro. Dans l'exemple de la figure 3.3a, après l'analyse du carré hachuré, le premier compteur prend la valeur 15 (nombre de carrés en traitillé+carré en gris). En utilisant un tel algorithme, on remarque que l'analyse se fait par taille de carré d'où le nom d'analyse hiérarchique.

3.2.2 Analyse descendante-ascendante cyclique:

L'analyse est effectuée ici du niveau global au niveau local de manière cyclique. Une approche récursive est présentée par l'organigramme de la figure 3.4.

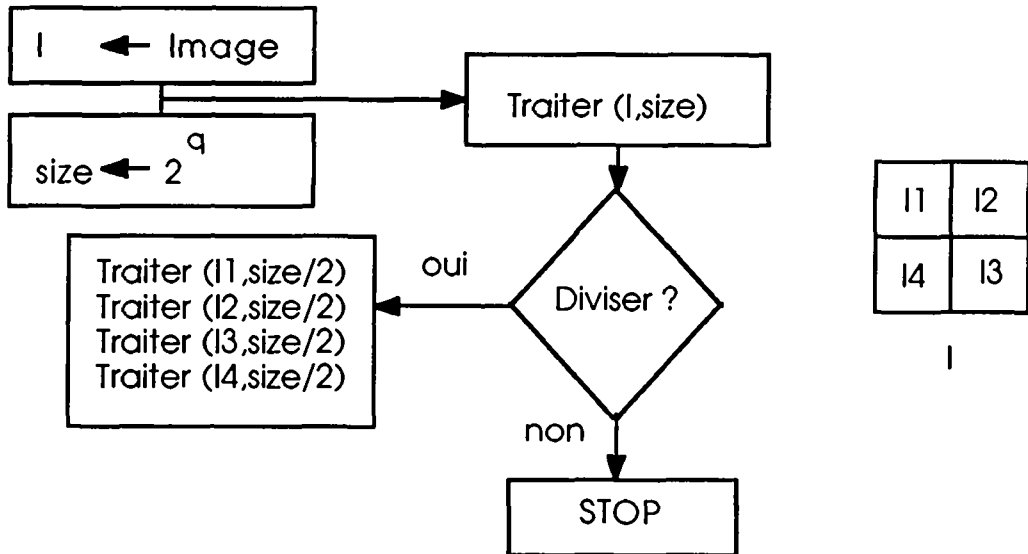


Figure 3.4: Implantation récursive de l'algorithme de division avec analyse descendante-ascendante cyclique.

Sans faire appel à la notion de récursivité, on peut définir une implantation par analyse cyclique en utilisant la structure de liste du paragraphe 3.2.1. Il suffit pour cela d'insérer dans la liste les coordonnées des carrés de taille immédiatement inférieure *juste après la position courante* du carré analysé. Dans ce cas, on n'a plus besoin de mémoriser de compteurs puisque l'on sait qu'un découpage entraînera seulement une insertion de quatre carrés de taille immédiatement inférieure. On peut également appliquer cet algorithme tel quel à des images de dimension quelconque. Il suffit simplement de modifier la liste en conséquence avant de commencer l'analyse.

Nous verrons à la section suivante que les opérations réalisées à un même niveau peuvent être grandement simplifiées par un choix adéquat de l'origine. Cette considération a justifié le choix d'une analyse hiérarchique dans la réalisation pratique de l'algorithme de division. Une fois le mode d'implantation choisi, il s'agit de décrire dans ce qui suit la complexité de chaque phase de l'algorithme.

3.3 Complexité de l'algorithme de division :

La complexité est liée principalement aux calculs de l'approximation optimale au sens des moindres carrés. Nous verrons comment la forme carrée des régions après division ainsi que le choix de fonctions de base polynomiales permettent de simplifier considérablement le problème. Il sera également montré comment cette

phase de la segmentation peut être traitée à partir de fonctions polynomiales séparables.

Des considérations sur les images naturelles traitées permettront de limiter le nombre de carrés à étudier pour aboutir à la partition après division et de simplifier les critères d'erreur choisis.

3.3.1 Aspects liés à l'approximation:

Le premier point à traiter consiste à choisir l'origine de l'espace bidimensionnel par rapport à laquelle seront effectués tous les calculs. Il est apparu avantageux de fixer une origine par rapport à chaque carré tout au long de l'opération de division. En effet, si l'on part de l'expression (2.15) qui représente le système à résoudre pour le calcul optimal des coefficients, on remarque que l'expression des matrices \mathbf{Z} et par conséquent \mathbf{S} n'est liée qu'aux positions des points du domaine D et non pas à la valeur du signal en ces points. Deux domaines, dont l'un peut être obtenu par translation géométrique de l'autre, auront donc les mêmes matrices \mathbf{Z} et \mathbf{S} associées si l'on choisit une origine identique relativement à chaque domaine (par exemple, son centre de symétrie). Ainsi, on a réduit le calcul de l'approximation pour chaque région à l'estimation du vecteur \mathbf{h} et à la résolution du système $\mathbf{S}\mathbf{u} = \mathbf{h}$, ce qui revient à $O(\max(Nr, r^3))$ opérations, où N représente le nombre de points du domaine et r le nombre de fonctions de base. Ce principe s'applique parfaitement à l'algorithme de division vu qu'à chaque niveau, on traite des carrés de même taille.

Pour chaque niveau de division, on n'estimera qu'une fois l'expression des matrices \mathbf{S} et \mathbf{Z} . Celles-ci seront donc évaluées au plus q fois tout au long de l'algorithme (l'image a une taille initiale $2^q \times 2^q$).

Le deuxième point est lié à l'expression de la matrice \mathbf{S} si l'on utilise des fonctions polynomiales pour approximer les données de départ. L'annexe C montre que chaque terme de cette matrice peut être exprimé par un produit de deux séries de coordonnées élevées à une certaine puissance (voir (c1)). Appelons (k_0, l_0) les coordonnées de l'origine relative à un carré de taille donnée. D'une façon générale, chaque série associée à la coordonnée k sera représentée par:

$$\sum_{k=1}^K (k-k_0)^m = \sum_{k=1}^K \left(\sum_{n=0}^m \binom{m}{n} k^n \cdot k_0^{m-n} \right) \quad (3.2)$$

où m représente un nombre entier positif et K correspond au nombre de points sur le côté du carré, c'est-à-dire une puissance de 2. En intervertissant les deux opérateurs de somme, on obtient :

$$\sum_{k=1}^K (k-k_0)^m = \sum_{n=0}^m \binom{m}{n} k_0^{m-n} \cdot \sum_{k=1}^K k^n \quad (3.3)$$

Une expression semblable est obtenue en remplaçant k et k_0 par l et l_0 respectivement pour la série relative à la coordonnée l . Dans la formule (3.3), on voit apparaître la somme de K entiers consécutifs élevés à la puissance m . La valeur de m la plus élevée apparaissant dans la matrice \mathbf{S} correspond à p^2 si p est le degré du polynôme considéré. En se limitant aux polynômes bidimensionnels de degré 3, on ne considèrera que les séries entières de puissances 0 à 6, notées par σ_0 à σ_6 . Leurs expressions sont données par [54, p.36]:

$$\begin{aligned} \sigma_0 &= K & \sigma_1 &= K(K+1)/2 & \sigma_2 &= \sigma_1(2K+1)/3 \\ \sigma_3 &= \sigma_1^2 & \sigma_4 &= \sigma_2(3K^2+3K-1)/5 & & (3.4) \\ \sigma_5 &= \sigma_3(2K^2+2K-1)/3 & \sigma_6 &= \sigma_2(3K^4+6K^3-3K+1)/7 \end{aligned}$$

A partir de ces relations, il est très simple d'évaluer l'expression (3.3). On peut simplifier ultérieurement le calcul de la matrice \mathbf{S} en choisissant pour origine le centre de symétrie du carré analysé. En effet, dans ce cas, toutes les séries correspondant à des valeurs impaires de m sont nulles (voir annexe C). La matrice \mathbf{S} prend alors l'expression (c2) pour des polynômes de degré 3 et se limite aux sous-matrices de taille 1x1, 3x3 et 6x6 inscrites dans la partie supérieure gauche de la matrice représentée par (c2) pour des polynômes de degré 0, 1 et 2, respectivement.

Les termes a , b , c et d de l'annexe C s'estiment pour un carré de taille donnée à partir des relations (3.3).et (3.4). En utilisant un programme de manipulations d'expressions symboliques, il est encore possible de faciliter la tâche de la résolution du système (2.15) en explicitant l'expression des inverses des matrices \mathbf{S} . Appelons \mathbf{S}_0 , \mathbf{S}_1 , \mathbf{S}_2 et \mathbf{S}_3 , les matrices correspondant à chaque degré de polynôme. On peut écrire:

$$\mathbf{S}_0^{-1} = (1/K^2) \quad (3.5)$$

$$\mathbf{S}_1^{-1} = \begin{bmatrix} 1/a^2 & 0 & 0 \\ 0 & 1/(ab) & 0 \\ 0 & 0 & 1/(ab) \end{bmatrix} \quad (3.6)$$

$$\mathbf{S}_2^{-1} = \begin{bmatrix} \alpha_2 & 0 & 0 & \beta_2 & 0 & \beta_2 \\ 0 & \gamma_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma_2 & 0 & 0 & 0 \\ \beta_2 & 0 & 0 & \delta_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/b^2 & 0 \\ \beta_2 & 0 & 0 & 0 & 0 & \delta_2 \end{bmatrix} \quad (3.7)$$

où $\delta_2 = 1/(ac-b^2)$, $\alpha_2 = (ac+b^2)/(a^2\delta_2)$, $\beta_2 = -b/(a\delta_2)$ et $\gamma_2 = 1/(ab)$.

$$S_3^{-1} = \begin{bmatrix} \alpha_3 & 0 & 0 & \beta_3 & 0 & \beta_3 & 0 & 0 & 0 & 0 \\ 0 & \gamma_3 & 0 & 0 & 0 & 0 & \delta_3 & 0 & \varepsilon_3 & 0 \\ 0 & 0 & \gamma_3 & 0 & 0 & 0 & 0 & \varepsilon_3 & 0 & \delta_3 \\ \beta_3 & 0 & 0 & -\varepsilon_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/b^2 & 0 & 0 & 0 & 0 & 0 \\ \beta_3 & 0 & 0 & 0 & 0 & -\varepsilon_3 & 0 & 0 & 0 & 0 \\ 0 & \delta_3 & 0 & 0 & 0 & 0 & \eta_3 & 0 & 0 & 0 \\ 0 & 0 & \varepsilon_3 & 0 & 0 & 0 & 0 & \kappa_3 & 0 & 0 \\ 0 & \varepsilon_3 & 0 & 0 & 0 & 0 & 0 & 0 & \kappa_3 & 0 \\ 0 & 0 & \delta_3 & 0 & 0 & 0 & 0 & 0 & 0 & \eta_3 \end{bmatrix} \quad (3.8)$$

où $\alpha_3 = \alpha_2$, $\beta_3 = \beta_2$, $\delta_3 = c/(a(c^2 - bd))$, $\varepsilon_3 = -\delta_2$, $\gamma_3 = \delta_2 \delta_3 (ad - bc)$,
 $\eta_3 = b/(a(bd - c^2))$ et $\kappa_3 = a\delta_2/b$.

Comme l'expression des matrices ci-dessus reste inchangée pour toute image traitée, il suffit de les mémoriser pour toutes les tailles possibles de carrés. Pour justifier le gain en calcul qui peut ainsi être obtenu, le tableau 3.5 compare le nombre d'opérations nécessaires pour des polynômes de différents degrés sur une certaine image. Prenons par exemple, l'image du caméraman. L'estimation reste valable pour d'autres catégories d'images car on peut observer expérimentalement que le nombre de carrés de chaque taille est du même ordre de grandeur d'une image à l'autre. Après application de l'algorithme de division en utilisant un polynôme de degré 3, l'image du caméraman est formée d'un carré de taille 64x64, de 11, 33, 130 et 2088 carrés de taille 32x32, 16x16, 8x8 et 4x4 respectivement. Pour arriver à ce résultat, il a fallu analyser 16, 60, 196, 652 et 2088 carrés de tailles respectives 64x64, 32x32, 16x16, 8x8 et 4x4, en supposant qu'on ait commencé l'algorithme de division à partir de 16 carrés 64x64. Une estimation de la complexité de calcul devient dès lors possible. Un raisonnement similaire peut être utilisé pour des polynômes de degré inférieur. Le cas des polynômes de degré 0 ne sera pas pris en considération ici, car des simplifications ultérieures de l'algorithme peuvent être utilisées. Dans l'estimation de la complexité pour une origine placée au centre de symétrie de chaque carré (origine locale), on a utilisé les relations (3.6) à (3.8) pour le calcul du produit $S^{-1}h$, ce qui peut être obtenu en environ 3, 10 et 22 opérations pour des polynômes de degré 1, 2 et 3 respectivement. Si l'on ne considère qu'une origine unique (origine globale), le calcul de la complexité est alors réduit aux étapes 2 à 4 décrites au paragraphe 2.3.1.

m \	Origine locale	Origine globale
1	840	2 770
2	1530	7510
3	2 560	19 450

Tableau 3.5: Estimation de la complexité de calcul de l'algorithme de division. Les valeurs indiquées sont en milliers d'opérations nécessaires. (m=degré du polynôme).

Rôle de la séparabilité:

Il est possible de limiter la complexité de l'algorithme de division par le choix d'un ensemble de fonctions polynomiales séparables. Il est en réalité plus correct de dire que la solution du problème de l'approximation au sens des moindres carrés est rendue séparable pour des régions de forme rectangulaire par ce choix adéquat de fonctions de base. D'une façon générale, de tels polynômes sont décrits par la relation:

$$g^{\wedge}(x,y) = \sum_{i=1}^{r_1} \sum_{j=1}^{r_2} u_{ij} x^{i-1} y^{j-1} \quad (3.9)$$

Si l'on désire approximer une région de forme rectangulaire constituée par un ensemble de K lignes et L colonnes, la région approximée qui peut être notée par une matrice G^{\wedge} de taille KxL, pourra être définie au moyen de la relation:

$$G^{\wedge} = V_k \cdot U \cdot V_l^T \quad (3.10)$$

où V_k et V_l sont des pseudo-matrices de Vandermonde de taille respective $K \times r_1$ et $L \times r_2$. Chaque ligne de coordonnée x d'une pseudo-matrice de Vandermonde V de taille $N \times r$ s'exprime par le vecteur $[1 \ x \ x^2 \ \dots \ x^{r-1}]$. La somme des erreurs quadratiques d'approximation peut-être exprimée par:

$$\begin{aligned} EQ &= \sum_{k=1}^K \sum_{l=1}^L (g(k,l) - g^{\wedge}(k,l))^2 \\ &= \text{Trace} ((G - G^{\wedge})^T \cdot (G - G^{\wedge})) \end{aligned} \quad (3.11)$$

où \mathbf{G} représente le tableau des données originales. La solution qui minimise cette erreur est obtenue par annulation de la dérivée $\partial EQ/\partial \mathbf{U}$. Elle satisfait l'équation [55]:

$$\mathbf{U} = (\mathbf{V}_k^T \mathbf{V}_k)^{-1} \cdot \mathbf{V}_k^T \cdot \mathbf{G} \cdot \mathbf{V}_1 \cdot (\mathbf{V}_1^T \mathbf{V}_1)^{-1} \quad (3.12)$$

Pour simplifier la notation, nous appellerons \mathbf{S}_k la matrice carrée de taille $r_1 \times r_1$ $\mathbf{V}_k^T \mathbf{V}_k$; \mathbf{S}_1 la matrice carrée de taille $r_1 \times r_1$ $\mathbf{V}_1^T \mathbf{V}_1$; enfin, la matrice $\mathbf{V}_k^T \cdot \mathbf{G} \cdot \mathbf{V}_1$ de taille $r_1 \times r_2$ sera appelée \mathbf{H} . On remarque que les matrices \mathbf{S}_k et \mathbf{S}_1 sont de type Hankel. En effet, elles ont la forme:

$$\begin{bmatrix} v_0 & v_1 & \dots & v_{r-1} \\ v_1 & v_2 & \dots & v_r \\ | & | & \dots & | \\ v_{r-1} & v_r & \dots & v_{2r} \end{bmatrix} \quad (3.13)$$

Cette matrice est entièrement spécifiée par $2r+1$ termes, qui dans le cas des fonctions polynomiales choisies s'expriment pour \mathbf{S}_k par:

$$v_i = \sum_{k=1}^K k^i \quad (3.14)$$

La résolution du système (3.12) est beaucoup plus facile à obtenir que dans le cas d'un polynôme de forme générale. En effet, celle-ci nécessite:

1.- l'inversion de deux matrices de Hankel de tailles respectives $r_1 \times r_1$ et $r_2 \times r_2$, ce qui peut être effectué en $O(r_1^2)$ et $O(r_2^2)$ opérations au moyen d'algorithmes appropriés [56].

2.- cinq multiplications matricielles. Si l'on considère des carrés de taille $t \times t$, ces multiplications peuvent être obtenues en $\min(r_1, r_2) \times t^2 + r_1 \times r_2 \times t + r_1^2 \times r_2 + r_1 \times r_2^2$ opérations, pour autant que celles-ci soient effectuées dans un ordre correct. Dès que le produit $\min(r_1, r_2)$ est plus grand que 1, cette approche est favorable par rapport à la formulation non séparable quelle que soit la valeur de t . Ceci est montré dans l'annexe D.

Pour ce qui est de l'expression de l'erreur quadratique, celle-ci peut également être obtenue sans effectuer le calcul complet des valeurs approximées représentées par la matrice \mathbf{G}^\wedge . Elle s'exprime par (voir annexe E):

$$EQ = \text{Trace}(\mathbf{G}^T \mathbf{G}) - \text{Trace}(\mathbf{U}^T \mathbf{H}) \quad (3.15)$$

Elle peut s'obtenir pour un carré de taille $t \times t$ en environ $t^2 + \min(r_1 + r_2^2, r_1^2 + r_2)$ opérations (voir annexe E). Cette approche séparable s'avère de nouveau plus efficace dès que r_1 et r_2 sont plus grands que 1.

Dans le contexte de l'algorithme de division, les considérations relatives au choix de l'origine sont indépendantes du choix des fonctions de base et s'appliquent donc parfaitement aux polynômes séparables. En plaçant celle-ci au centre de symétrie de chaque carré, on rend bon nombre d'éléments des matrices de Hankel S_k et S_l nuls (tous les termes correspondant à des valeurs de i impaires). Ceci permet de simplifier une nouvelle fois la complexité de l'algorithme de division en limitant les calculs liés aux approximations moindres carrés.

Le tableau 3.6 compare pour différentes valeurs de $r_1 \times r_2$ et t la complexité de calcul liée à une formulation séparable et non séparable du problème d'approximation. On a utilisé les expressions de N_s et N_{ns} données dans l'annexe E, où l'on a négligé les opérations relatives aux inversions de matrice. En effet, celles-ci peuvent être considérablement réduites par un choix adéquat de l'origine comme centre de symétrie de chaque carré.

taille poly	t = 4	t = 8	t = 16	t = 32	t = 64
$r_1 = 2$ $r_2 = 2$	64	176	592	2192	8464
	80	272	1040	4112	16400
$r_1 = 2$ $r_2 = 3$	86	206	638	2270	8606
	132	420	1572	6180	24612
$r_1 = 2$ $r_2 = 4$	112	240	688	2352	8752
	192	576	2112	8256	32832
$r_1 = 3$ $r_2 = 3$	138	318	966	3414	12918
	225	657	2385	9297	36945
$r_1 = 3$ $r_2 = 4$	180	372	1044	3540	13140
	336	912	3216	12432	49296

Tableau 3.6: Comparaison du nombre d'opérations nécessaires au calcul de l'approximation polynomiale séparable entre une formulation séparable et non séparable du problème d'approximation. Pour chaque valeur des paramètres, la ligne supérieure correspond à N_s , la ligne inférieure à N_{ns} .

3.3.2 Simplifications rendues possibles grâce à la structure des images traitées:

Les images traitées sont supposées contenir de l'information même détaillée. Il est donc raisonnable de penser qu'une scène d'intérêt apparaît fréquemment au centre de l'image, vu que l'on tâche de cadrer le plus souvent la scène. L'application de l'algorithme de division ne peut en général pas donner de carré de taille supérieure à 64×64 pour une image de départ de taille 256×256 . Ceci justifie de commencer l'application de l'algorithme de division à un niveau relativement élevé, d'autant plus qu'on utilise un nombre de coefficients très restreint pour

reconstituer le signal sur un grand domaine. Aussi, on a choisi de faire débiter l'analyse avec des carrés de taille 64x64.

Il apparaît vraisemblable que pour chaque carré de grande taille analysé au cours du processus de division, l'erreur de contour (EC) soit systématiquement plus grande que le seuil T_2 que l'on a fixé pour éviter de perdre de l'information, ceci à cause de l'impossibilité de représenter un saut de luminance au moyen d'une approximation polynomiale bidimensionnelle calculée selon le critère des moindres carrés lorsque le nombre de points du domaine analysé est beaucoup plus grand que le nombre de coefficients du polynôme. Une subdivision du carré en question est dès lors jugée nécessaire. On peut donc choisir de mesurer simplement la présence d'un contour, pour décider s'il faut subdiviser ultérieurement le carré. Le calcul de l'approximation optimale puis de l'erreur quadratique moyenne (EQM) n'est alors effectué que si le carré analysé ne présente pas de contours. Ceci permet de juger de la correspondance moyenne entre les données de départ et celles approximées. Pour les petites tailles de carré, c'est-à-dire lorsque le nombre de pixels est sensiblement égal au nombre de coefficients du polynôme, on peut vérifier qu'un contour contenu à l'intérieur du carré en question peut être correctement reproduit. Dans ce cas, il est également inutile de mesurer l'erreur de contour (EC) car elle s'apparente relativement bien à l'erreur quadratique moyenne (la plupart des points du carré correspondant à des points situés de part et d'autre du contour). Grâce à ces considérations, il est ainsi possible d'éviter bon nombre de calculs d'approximation et d'éliminer la mesure d'erreur de contour EC. L'algorithme de division de la figure 3.2 peut donc se simplifier comme indiqué à la figure 3.7 sans distorsion complémentaire.

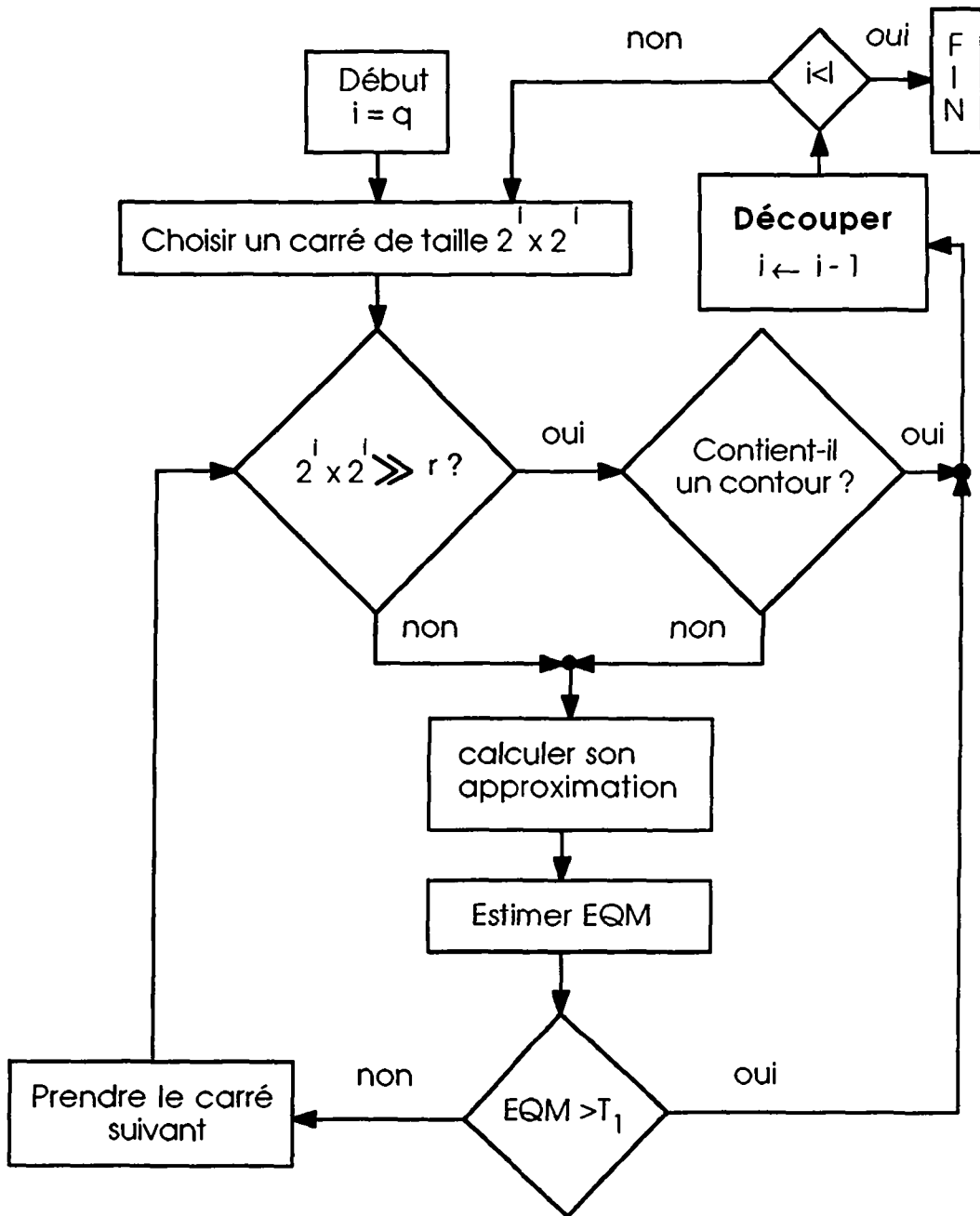


Figure 3.7: Algorithme de division simplifié par analyse de présence des contours (l: dernier niveau de division défini par la relation (3.1)).

Une fois l'image analysée au moyen de l'algorithme de division, nous allons étudier dans la section qui suit comment représenter la segmentation en carrés ainsi obtenue.

3.4 Représentation de l'information segmentée et postraitement :

Si l'on considère la segmentation obtenue après application de l'algorithme de division, on peut reconstruire une image approximée en connaissant la structure de l'image divisée et les coefficients d'approximation polynomiaux pour chaque carré. Le premier paragraphe illustre comment représenter l'information de position, le second indique comment il est possible de quantifier l'information relative aux coefficients polynomiaux. Ceci permettra d'estimer la compression que l'on peut atteindre à ce niveau du traitement de l'algorithme de division et rassemblement. Enfin, le dernier paragraphe décrit le genre de postraitements qui peuvent être utilisés de manière à rehausser la qualité de l'image reconstruite après application de l'algorithme de division.

3.4.1 Position et taille des carrés:

Dans ce paragraphe, deux méthodes sont proposées pour coder la position et la taille des carrés obtenus par l'algorithme de division. La première est basée sur la structure d'arbre quaternaire qui peut être utilisée pour représenter la partition de l'image en carrés. La seconde est un peu plus complexe mais permet de générer la partition avec un nombre plus réduit de bits.

Codage par arbre quaternaire:

Le graphe de la segmentation représentant l'image divisée peut se ramener à un arbre quaternaire. La figure 3.8b illustre la représentation en arbre quaternaire de la partition de la figure 3.8a.

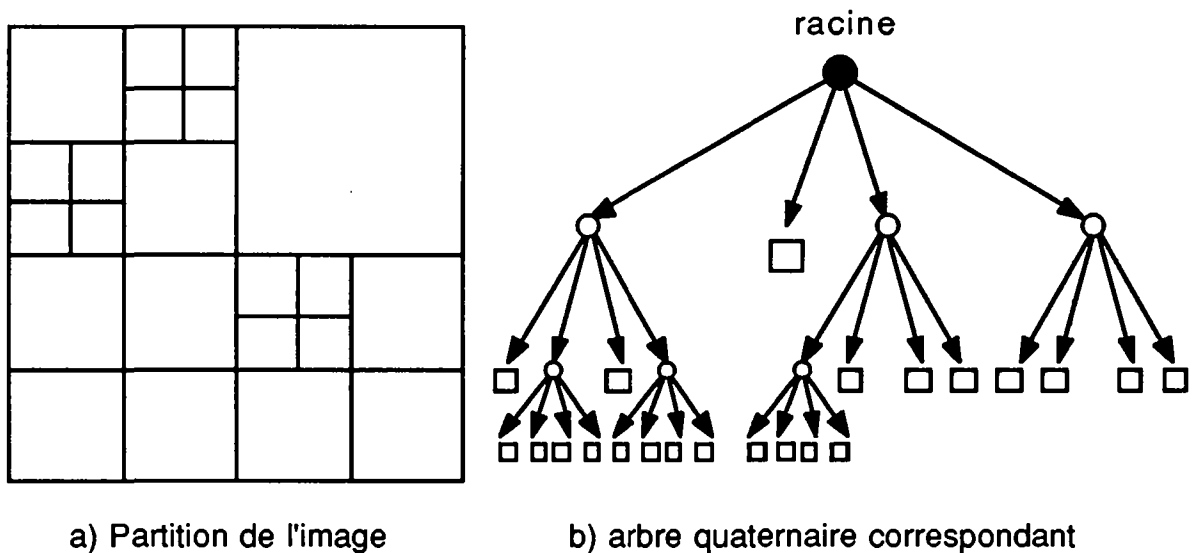


Figure 3.8: Représentation par arbre quaternaire.

Les arbres quaternaires définissent une structure de données hiérarchique compacte. Leur utilisation n'est pas forcément liée à la segmentation d'images [57,25]. Dans le cadre des applications pour la compression d'images, ils ont l'avantage de permettre une transmission progressive de l'information. Récemment, Cohen et al. [58] ont analysé, au moyen d'arbres quaternaires, la compression que l'on pouvait atteindre pour le codage hiérarchique d'images binaires.

Le principe de codage qui peut être obtenu pour représenter la structure des carrés obtenus après division est relativement simple. En partant de la racine de l'arbre, on transmet un bit pour dire si l'on peut descendre à un niveau plus bas, ou si l'on se trouve sur une feuille de l'arbre. Une autre manière de voir les choses consiste à transmettre un bit pour déterminer s'il est nécessaire de diviser le carré analysé ou pas. On peut éviter de transmettre le bit correspondant au dernier niveau de l'arbre si l'on sait qu'une subdivision ultérieure est impossible (taille du pixel atteinte ou nombre de pixels du carré immédiatement supérieur au nombre de coefficients d'approximation). Bien entendu, ceci présuppose que le récepteur connaît la taille de l'image à transmettre et le nombre de coefficients d'approximation. De la même façon, on peut commencer le code à un certain niveau dans l'arbre, si l'on sait que l'image ne peut présenter de carrés de taille plus grande que 64×64 (cf. paragraphe 3.3.2). De nouveau, on suppose que cette information est connue du récepteur.

Le coût le plus élevé que l'on puisse obtenir avec un tel schéma de codage correspond à une partition constituée uniquement de carrés de taille la plus petite possible, soit $2^l \times 2^l$. Le coût le moins élevé consiste à avoir des carrés de taille 64×64 uniquement. En appelant C un tel coût exprimé en bits, on a la relation (en supposant $q \geq 6$):

$$2^{2(q-6)} \leq C \leq \sum_{i=l+1}^q 2^{2i} = (4^{q+1} - 4^{l+1})/3 \quad (3.16)$$

La section suivante illustrera pratiquement le coût nécessaire pour les trois images traitées selon divers degrés du polynôme. On remarquera que le coût est souvent proche de la borne supérieure car le nombre de carrés de taille $2^l \times 2^l$ est très élevé.

Ce code même s'il permet une transmission hiérarchique, par taille de carré, ne s'adapte pas bien à des images rectangulaires de dimension quelconque. En outre, il utilise un faible nombre de bits pour transmettre un carré de grande taille. Or de tels carrés apparaissent rarement après application de l'algorithme de division. Ceci va à l'encontre de la philosophie de coût minimum qui tend à associer un mot long à un symbole rare et inversement. Néanmoins, ce code tient efficacement compte des contraintes géométriques imposées par l'algorithme de division et par la taille de l'image traitée. Pour tenir compte de toutes ces observations, on a défini un code un peu plus efficace sans que sa complexité en soit sensiblement accrue. Chaque fois qu'un carré est transmis, les contraintes géométriques de l'algorithme de division permettent de modifier la table de correspondance de chaque symbole utilisé.

Codage conditionnel par taille de carré:

Dans cette approche, on choisit pour symbole la taille de chaque carré. En effet, en transmettant successivement la taille de chaque carré selon un ordre de balayage prédéfini, il est possible de retrouver la partition donnée par l'algorithme de division. Par la suite, on supposera que le sens de parcours ira de gauche à droite et de haut en bas.

De manière à trouver un code efficace pour chacun de ces symboles, on estime leur probabilité d'apparition p_6 à p_1 en supposant que les carrés ont des tailles allant de $2^1 \times 2^1$ à 64×64 ainsi qu'on en a fait l'hypothèse au paragraphe précédent. Il existe donc 7-1 symboles à coder que l'on notera 1 à 6. Pour la plupart des images réelles testées, le nombre de carrés de taille $2^i \times 2^i$ est beaucoup plus faible que le nombre de carrés $2^{i-1} \times 2^{i-1}$. On peut dès lors admettre que la probabilité d'apparition de n'importe quel symbole correspondant à des carrés de taille $2^i \times 2^i$ à 64×64 ($\sum_{n=i..6} p_n$) reste plus faible que la probabilité d'apparition du symbole $2^{i-1} \times 2^{i-1}$. Le code optimal de Huffman utilisant un alphabet binaire associera 1 bit au au symbole 1, 2 bits au symbole 1+1, ... , 5-1 bits au symbole 4, 6-1 bits aux symboles 5 et 6.

Pour améliorer les performances en compression, il suffit de tenir compte du fait que l'on ne peut pas avoir n'importe quelle séquence de symboles 1 à 6 à cause des contraintes géométriques imposées par la structure de l'image et l'algorithme de division. En effet, supposons d'avoir transmis le symbole 5 correspondant à un carré de taille 32×32 . S'il est inscrit dans la partie supérieure gauche d'un carré de plus grande taille (64×64) analysé au cours de l'application de l'algorithme, on peut en déduire que tous les carrés qui lui sont adjacents à sa droite et au-dessous de lui ainsi que selon la diagonale descendante selon une mesure de contiguïté 8-connexes ne peuvent avoir qu'une taille inférieure ou égale à lui-même, en l'occurrence 32×32 . D'une façon plus générale, considérons la figure 3.9.

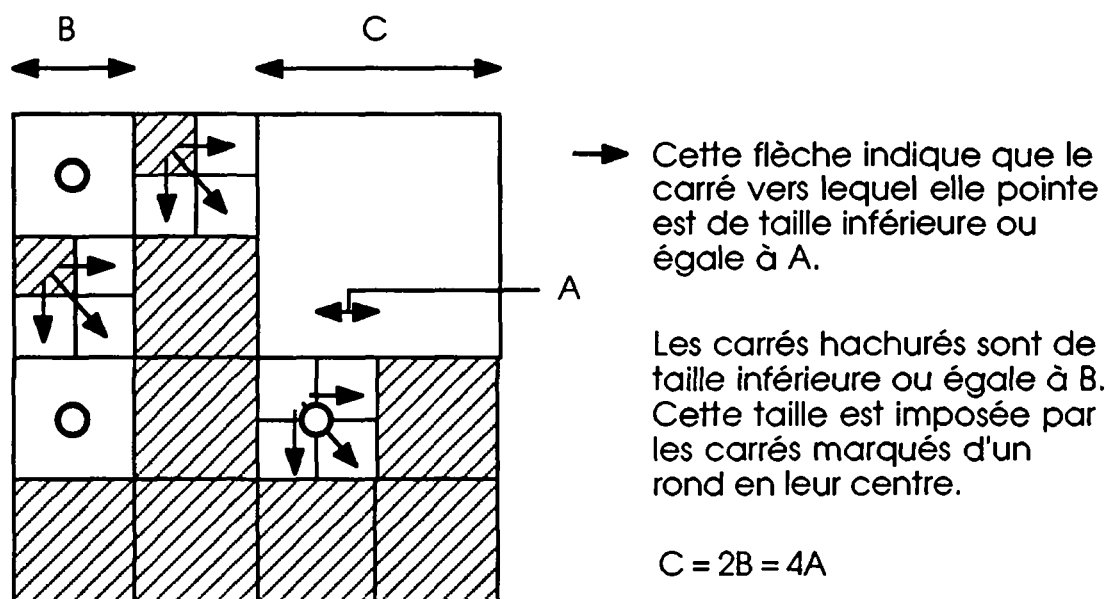


Figure 3.9: Contrainte géométrique imposée par l'algorithme de division.

Comme la contrainte géométrique imposée par l'algorithme de division agit selon une mesure de voisinage 8-connexes, on considèrera cette mesure de connexité dans ce paragraphe. Supposons vouloir transmettre la taille d'un certain carré. Regardons qu'elle est la taille que nous notons t du plus petit carré situé à sa gauche et au dessus de lui ainsi que selon la diagonale ascendante selon la mesure de contiguïté 8-connexes. Par exemple, dans la figure 3.9, le premier carré hachuré a pour voisin de gauche un carré de taille t égale à B. La structure de l'algorithme de division impose pour ce carré hachuré de ne pas avoir de taille supérieure à B. Ceci n'est valable que parce que le carré hachuré est inscrit dans

un carré de taille C égale à $2B$, analysé lors de l'application de l'algorithme de division. D'une façon générale, le carré courant est de taille inférieure ou égale à t s'il est inscrit dans un carré de taille $2t$ analysé lors de l'algorithme de division. En particulier, dans le cas du symbole l (carré de taille $2^l \times 2^l$), cela veut dire que les 3 symboles représentant ses voisins à sa droite, au-dessous de lui ainsi que selon la diagonale descendante selon une mesure de connexité d'ordre 8 seront égaux à l .

La contrainte géométrique peut être exprimée différemment en reprenant la représentation de l'algorithme de division par arbre quaternaire. La transmission de la taille du premier carré en haut à gauche de l'image impose quel est le niveau de la branche la plus à gauche de l'arbre. Dès lors, il existe 3 noeuds de même niveau à la droite de cette branche. Il suffit de transmettre si ces noeuds correspondent à des feuilles de l'arbre ou s'ils donnent naissance à de futures branches aboutissant à des noeuds de niveau inférieur. On parcourt ainsi l'arbre de gauche à droite en transmettant le niveau de la prochaine branche.

On peut donc admettre de réduire la table de correspondance entre symboles à coder et mots binaires, selon les symboles déjà transmis. Si l'on veut connaître la taille d'un certain carré, sachant qu'il est forcément de taille inférieure ou égale à $2^i \times 2^i$ du fait des contraintes géométriques imposées par l'algorithme de division, le symbole à transmettre utilisera une table de correspondance identique à la précédente jusqu'à $i-1$ et utilisant le même nombre de bits que pour $i-1$ pour le symbole i . A la limite si i est égal à 1, aucun bit ne sera transmis et si i est égal à 6, la table restera inchangée. Ceci est illustré à la figure 3.10. Dans cette figure, les carrés blancs, gris foncé et gris clair utilisent une table de correspondance dont le mot binaire le plus long est de 4, 3 et 2 bits, respectivement.

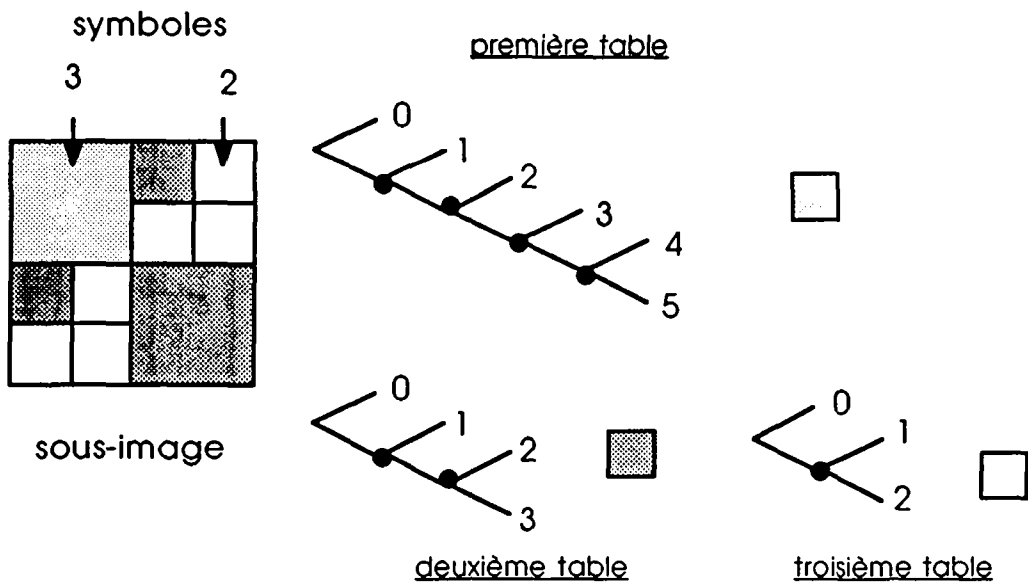


Figure 3.10: Changement de table de correspondance.
(le remplissage des carrés indique la table à utiliser)

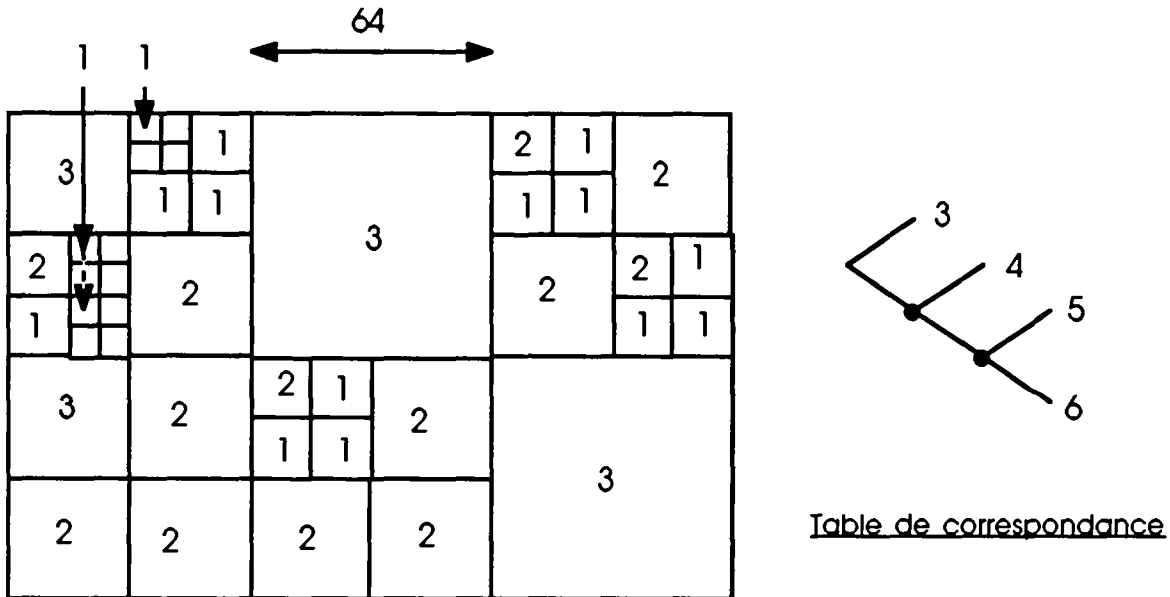
Selon la partition obtenue, il peut être avantageux de transmettre les différents symboles selon une autre direction de balayage tout en gardant la même philosophie de codage (gauche à droite+haut en bas; gauche à droite+bas en haut; droite à gauche+haut en bas; droite à gauche+bas en haut; haut en bas+gauche à droite...). Il existe huit possibilités correspondant chacune à un certain coût de

transmission. Il suffit de transmettre trois bits supplémentaires au récepteur pour indiquer la direction correspondant au coût le plus faible.

Le coût le plus élevé que l'on puisse obtenir avec ce schéma de codage correspond à une partition constituée uniquement de carrés de taille la plus petite possible, soit $2^l \times 2^l$, comme dans le cas du codage par arbre quaternaire, à cause des hypothèses faites sur l'apparition de chaque symbole. Le coût le moins élevé consiste aussi à avoir des carrés de taille 64×64 uniquement. En gardant la même notation qu'au paragraphe précédent, on a:

$$(6-l) \cdot 2^{2(q-6)} \leq C \leq 4(q-l-1) \quad (3.17)$$

Comme en général il y a beaucoup plus de carrés de taille $2^l \times 2^l$, il s'avère plus efficace que le codage par arbre quaternaire. Les exemples présentés à la section 3.5 confirmeront ceci dans le cas d'images réelles. La figure 3.11 illustre comment comptabiliser le nombre de bits nécessaires pour une certaine partition.



Partition de l'image (l=3)

Figure 3.11: Le nombre de bits nécessaire à la transmission de chaque carré de la partition ci-dessus est indiqué au centre du carré.

Pour trouver le coût nécessaire à la représentation de la partition, on compte le nombre de carrés de chaque taille correspondant aux branches les plus à gauche à partir de chaque noeud de l'arbre quaternaire associé à la partition (attention, si l'on utilise un arbre quaternaire pour représenter la partition de l'image, le code conditionnel par taille de carré est utilisé ici). Notons ainsi G_1 à G_6 ces nombres pour chaque taille possible de carré. Les nombres de carrés restant sont notés similairement N_1 à N_6 . Le coût est alors donné par la relation:

$$C = (0 \times N_1 + 1 \times G_1) + (1 \times N_{1+1} + 2 \times G_{1+1}) + \dots + ((5-l) \times N_5 + (6-l) \times G_5) + (6-l) \times (N_6 + G_6)$$

$$= G_6 - \sum_{i=1}^6 ((i-1) \times (N_i + G_i) + G_i) \quad (3.18)$$

Disons encore que le code conditionnel présente l'avantage de pouvoir s'appliquer à des images rectangulaires de taille quelconque.

3.4.2 Quantification des coefficients polynomiaux:

Ce thème sera traité en détail au chapitre 5 après l'opération de rassemblement. Il s'agit de donner dans ce paragraphe quelques indications qui permettront d'estimer d'une manière indicative la compression que l'on peut atteindre déjà à ce niveau de la segmentation.

Quelle que soit la forme de l'approximation, on montre justement dans ce chapitre que les coefficients d'approximation des polynômes peuvent être quantifiés avec au plus huit bits par coefficient en conduisant à une erreur de quantification négligeable. On admet que l'image originale a été quantifiée à huit bits.

En effet, si l'on a r coefficients d'approximation, il suffit de transmettre r pixels quantifiés à huit bits du carré approximé. On est alors en mesure de calculer les autres pixels du carré en interpolant les r pixels précédents au moyen des mêmes fonctions de base $\Psi_j(x,y)$. La seule erreur que l'on peut commettre provient de l'arrondissement des r valeurs approximées au niveau de quantification le plus proche selon la méthode présentée à la section 2.3.3. On peut garder la même philosophie et décider de quantifier chaque coefficient avec moins de bits, soit b ce nombre. Il faut dans ce cas veiller à choisir de manière adéquate les r pixels quantifiés à b bits là où les erreurs d'arrondi sont les plus faibles. Cette approche sera utilisée dans la présentation des résultats de la section suivante.

On n'a cependant absolument pas tenu compte de la statistique des vecteurs de coefficients \mathbf{u} sur l'ensemble des carrés de l'image. Or, il existe une redondance certaine de ces coefficients pour des carrés appartenant à un même espace régulier. Une approche pour éliminer cette redondance consiste à effectuer le rassemblement des carrés définissant un même espace régulier comme il est proposé au chapitre suivant. Si l'on ne désire pas changer la forme de la segmentation, on peut essayer de quantifier vectoriellement l'espace des vecteurs \mathbf{u} . Ceci sort toutefois du cadre de la recherche effectuée ici.

On se contentera néanmoins de trouver un code simple permettant de réduire cette redondance dans le cas d'une approximation d'ordre zéro. Ceci permet d'avoir une idée de la compression atteignable sans autre après la procédure de rassemblement.

Au lieu de coder la valeur moyenne de chaque carré représentant l'approximation optimale au sens des moindres carrés au moyen de la fonction de base $\Psi_1(x,y)=1$, on va utiliser un code appelé CDC (Code de différence de carrés). Dans ce code, on transmet la différence entre les valeurs moyennes de deux carrés adjacents en supposant que l'on parcourt l'image selon une certaine convention

(par exemple, de gauche à droite puis de bas en haut). La longueur des mots codes associés à chaque différence possible a été établie sur la base de la statistique de ces différences. On a utilisé pour ce faire le code optimal de Huffman. Les résultats correspondants sont présentés à la section suivante.

3.4.3 Postraitement:

En observant les images approximées, notamment si l'on tolère des erreurs importantes, on peut remarquer l'apparition de faux contours aux frontières de certains carrés. Ce défaut est lié au choix de fonctions polynomiales pour approximer les données de départ. En principe, l'attribution du statut de faux contour ne peut se faire que par la comparaison avec l'image de contrôle des contours. De tels artefacts ne peuvent cependant apparaître qu'à la frontière des carrés de grande taille. En effet, dans le cas de petits carrés, le polynôme va pratiquement interpoler les données de départ.

Pour éliminer ces faux contours correspondant à la présence d'une information haute fréquence locale, il suffit de filtrer passe-bas l'image dans le voisinage de ces sauts artificiels de luminance. Plus le carré sera grand, plus le filtrage devra être important et plus la zone où il sera effectué sera large. La largeur de la zone filtrée a été ainsi choisie à 2^{i-1} pixels pour un carré de taille $2^i \times 2^i$. Ce choix a été fait de manière rendre la dimension de la zone lissée une fonction croissante de la taille des carrés. Une taille plus faible que 2^{i-1} ne permet pas d'éliminer correctement les faux contours. Une taille plus grande risque de dégrader de façon trop importante le signal d'approximation polynomial estimé selon le critère des moindres carrés. Le filtre passe-bas est calculé de manière à minimiser les variations du gradient local estimé perpendiculairement au faux contour (voir section 5.3)

De manière à éviter de transmettre de l'information supplémentaire à un récepteur pour parvenir à déceler quels sont les faux contours, on admet de filtrer l'image dans le voisinage des frontières de tous les carrés de grande taille, l'apparition d'un contour réel se situant à la frontière exacte entre deux carrés de grande taille étant relativement peu probable. Pour ce qui est des frontières ne présentant pas de faux contours, l'effet du filtrage sera imperceptible vu ses caractéristiques basses fréquences. La taille minimale des carrés qui présentent de faux contours a été fixée 8×8 pour des polynômes de degré 2 ou 3, à 4×4 pour des polynômes de degré 0 ou 1. On évite ainsi le risque de lisser des contours réels qui ne peuvent apparaître que pour des carrés de dimension immédiatement inférieure aux tailles précédemment indiquées (voir algorithme de la figure 3.7).

La figure 3.12 présente l'image du couple rehaussée lorsqu'on utilise des plans pour approximer chaque carré. La figure 3.12a montre la forme de la partition (les carrés de taille 1×1 sont présentés en gris); la figure 3.12b montre l'image approximée; la figure 3.12c indique en noir les régions où est effectué le filtrage passe-bas de rehaussement; enfin la figure 3.12d montre l'image restaurée. Pour faire apparaître l'erreur maximum tolérée en chaque pixel, on exprime le seuil T_1 comme un nombre élevé au carré du fait de l'aspect quadratique de la mesure choisie. Cette remarque reste valable pour la présentation de tous les résultats expérimentaux de ce chapitre.



a) Image approximée

b) Partition de l'image (5290 carrés)



c) Zones filtrées passe-bas (en noir)

d) Image restaurée.

Figure 3.12: Posttraitement de l'image du couple
(approximation par polynôme de degré 1(plan), $T_1=15^2$).

3.5 Analyse expérimentale de l'algorithme de division :

Dans cette section, nous allons mesurer la qualité des résultats de la segmentation en carrés après application de l'algorithme de division. Dans un premier temps, on essaiera d'appliquer cet algorithme sans l'utilisation d'une image de contrôle des contours selon le schéma de principe de la figure 3.2. Dans

le deuxième paragraphe, on discute des résultats lorsqu'on applique le schéma de l'algorithme simplifié de la figure 3.7, où l'on fait usage cette fois d'une image de contrôle des contours. On y relève en particulier l'étude du rôle du seuil T_1 sur la mesure d'erreur quadratique moyenne, du degré du polynôme d'approximation et de la compression d'information qui peut être obtenue à ce niveau du traitement. Enfin le dernier point analysé dans cette section traite de la dégradation qui apparaît si l'on limite volontairement la plus petite taille de carré à un niveau supérieur à celui indiqué par la relation (3.1).

Convention:

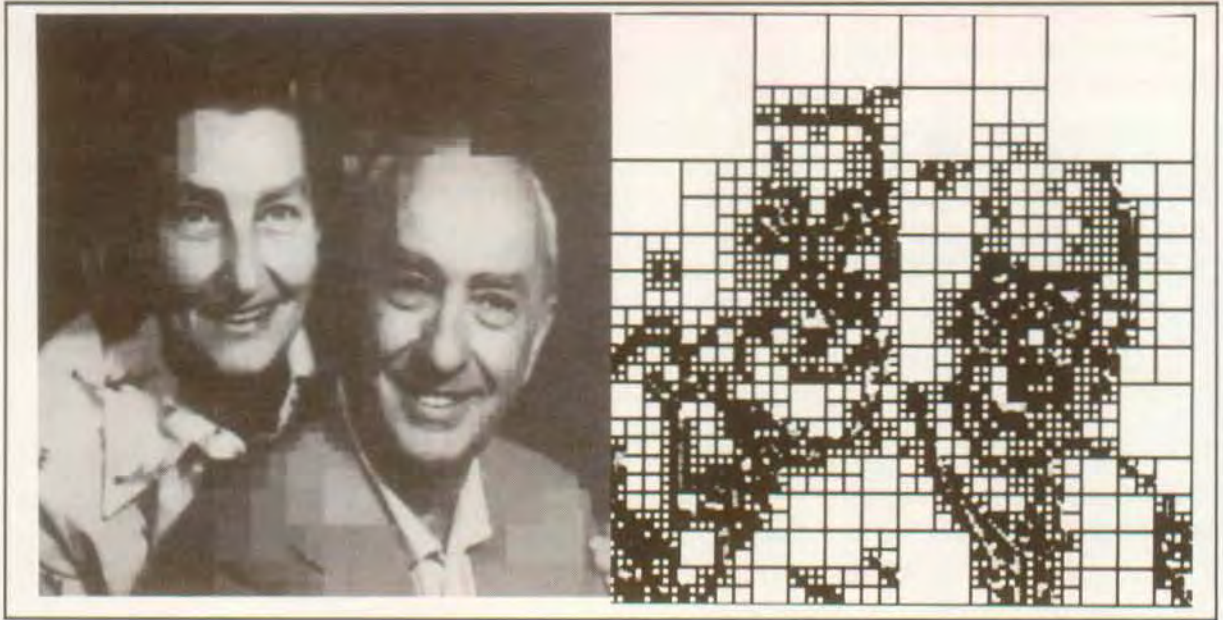
Les images présentées dans ce qui suit utilisent les notations suivantes. T_1 correspond au seuil sur l'erreur quadratique moyenne. α définit le degré du polynôme d'approximation. C_q et C_l indiquent les coûts nécessaires au codage de position des carrés lorsqu'on utilise le codage par arbre quaternaire et la codage conditionnel par taille de carré respectivement. ρ définit le rapport de compression sachant que l'on a codé chaque coefficient du polynôme avec 8 bits. Il s'exprime par la relation:

$$\rho = 256^2 \times 8 / (8rw + \min(C_q, C_l)) = 524288 / (8rw + \min(C_q, C_l)) \quad (3.19)$$

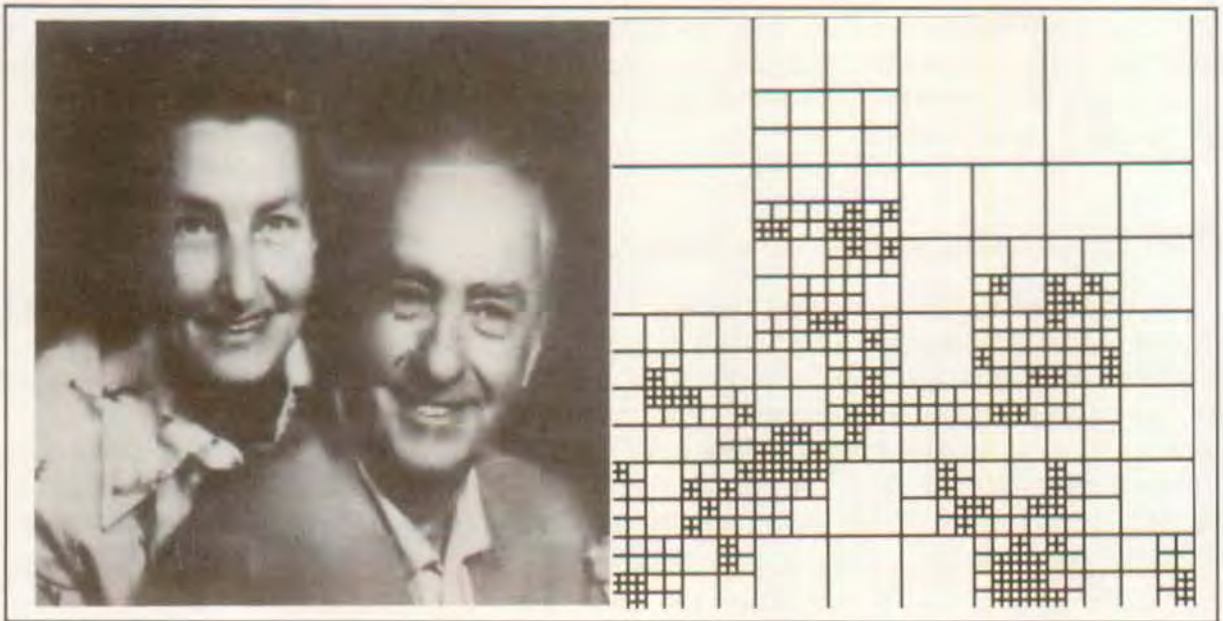
où r correspond au nombre de coefficients et w au nombre de carrés. Mis à part dans le paragraphe suivant, les images seront toujours divisées en utilisant une image de contrôle des contours calculée au moyen du filtre haute-fréquence présenté à la section 2.2.

3.5.1 Division effectuée sans image de contrôle des contours:

Les exemples qui sont présentés ici illustrent clairement la perte de l'information contour si l'on n'utilise qu'une mesure d'erreur quadratique moyenne comme seul critère d'appréciation de la qualité. Ceci reste indépendant du degré du polynôme choisi. La figure 3.13 montre les résultats obtenus pour l'image du couple. La première image indique la forme de la segmentation, la deuxième l'image reconstruite. Les degrés des polynômes utilisés dans les figures 3.13a, 3.13b, d'une part, 3.13c et 3.13d, d'autre part, sont 0 et 3 respectivement. Le seuil T_1 est fixé à 15^2 , le signal de départ étant quantifié à 255 niveaux disponibles. Les résultats peuvent être partiellement améliorés si l'on applique l'algorithme de rehaussement proposé au dernier paragraphe de la section précédente. Les images restaurées sont présentées à la figure 3.14. En gardant une même valeur de seuil pour T_1 , on présente à la figure 3.15 les résultats de l'algorithme de division pour l'image du bâtiment. Le degré du polynôme correspondant à 3.15a et 3.15b est fixé à 1 alors qu'il est de 2 pour 3.15c et 3.15d.

a) Image approximée ($\alpha=0, T_1=15^2$)b) Partition de l'image ($w=8317$).

$$C_q = 6056 \text{ bits} \quad C_l = 6618 \text{ bits} \quad \rho = 7.22.$$

c) Image approximée ($\alpha=3, T_1=15^2$)d) Partition de l'image ($w=583$).

$$C_q = 464 \text{ bits} \quad C_l = 522 \text{ bits} \quad \rho = 11.13.$$

Figure 3.13: Division de l'image du couple sans image de contrôle des contours.

Les images de la figure 3.13 mettent bien en évidence la perte de l'information contour que l'on a si l'on base le critère de division uniquement sur une mesure d'erreur quadratique moyenne. Ainsi les images des figures 3.13a et 3.13b font apparaître une perte des contours au voisinage des fronts des personnes ainsi que sur la veste de l'homme. Les carrés où l'on décèle ce genre de distorsion sont

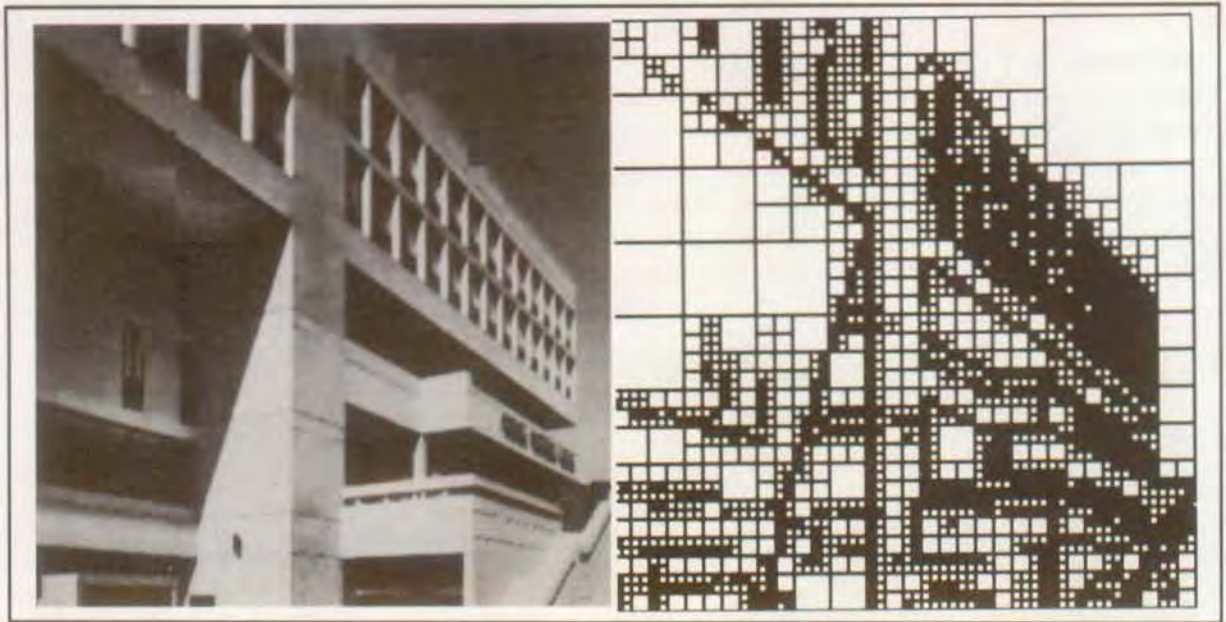
toujours de très grande taille (64x64,32x32). En effet, si le signal est bien approximé sur une large proportion du carré, l'erreur quadratique correspondante reste relativement faible. La contribution à cette erreur correspondant à la région située de l'autre côté du contour est rendue négligeable par la pondération par le nombre de points du carré. Lorsqu'on augmente le degré du polynôme à 3 (figures 3.13c et 3.13d), les distorsions sont encore plus visibles au niveau de la perte des contours réels de l'image. Par contre, on réduit le nombre des faux contours apparaissant à la jonction entre des carrés adjacents.



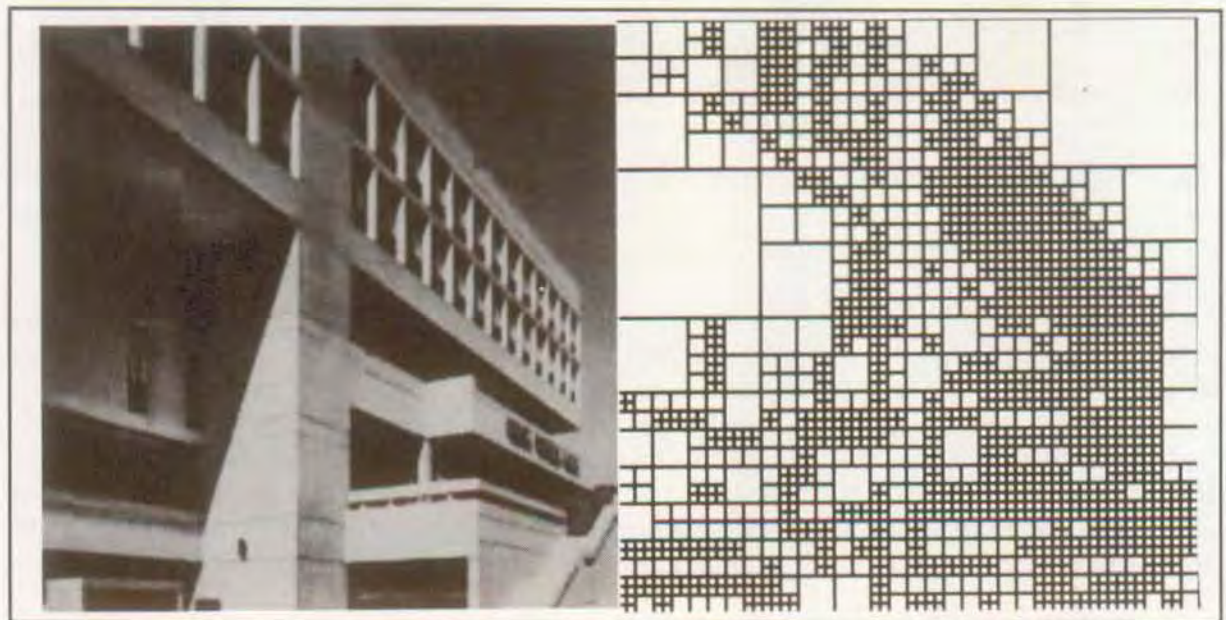
a) Image de la figure 3.13a rehaussée. b) Image de la figure 3.13c rehaussée

Figure 3.14: Images du couple de la figure 3.13 restaurées.

La figure 3.14 montre bien comment il est possible de réduire les distorsions dues à la perte des faux contours de l'image, en appliquant le filtre passe-bas sélectif par taille de carré proposé à la section 3.4.3. L'image de la figure 3.14b présente un aspect moins agréable à cause du faible nombre de carré de la partition obtenue au moyen de polynômes d'ordre 3. Il est important de remarquer que si ces dégradations peuvent être éliminées à ce niveau de la segmentation, il n'est plus possible d'obtenir les frontières réelles des contours après l'opération de rassemblement en partant d'un GCR basé sur les images segmentées de la figure 3.13.



a) Image approximée ($\alpha=1, T_1=15^2$) b) Partition de l'image ($w=4957$).
 $C_q = 2748$ bits $C_l = 2662$ bits $\rho = 4.31$.



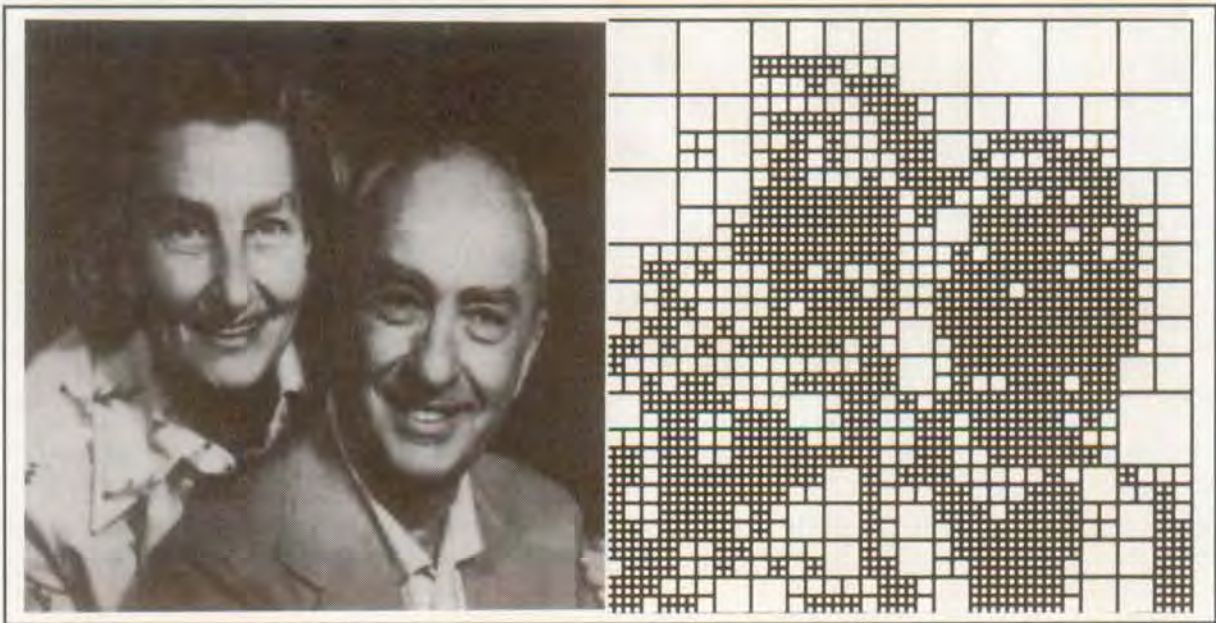
c) Image approximée ($\alpha=2, T_1=15^2$) d) Partition de l'image ($w=1825$).
 $C_q = 940$ bits. $C_l = 841$ bits $\rho = 5.93$.

Figure 3.15: Division de l'image du bâtiment sans image de contrôle des contours.

Les considérations effectuées à partir de la figure 3.13 peuvent être confirmées en observant les résultats de la figure 3.15. On remarque en plus que la perte de l'information contour est indépendante du degré du polynôme d'approximation. Disons encore que ce genre de dégradation apparaît presque toujours au voisinage des grandes régions homogènes de l'image (ciel pour

l'image du bâtiment, fond pour l'image du couple). Ceci résulte du choix du critère d'erreur globale. En gardant un même nombre de régions pour une même image, le critère d'erreur quadratique (EQ) aurait fait disparaître des régions homogènes de petite taille. Il aurait dès lors été impossible de rehausser la qualité de l'image par filtrage passe-bas local comme indiqué précédemment.

On peut éviter cependant de perdre l'information contour en plaçant le seuil T_1 beaucoup plus bas. Néanmoins, il en résulte une subdivision excessive de l'image de départ, certains carrés correspondant à des parties d'espace régulier ayant été découpés outre mesure. L'algorithme de rassemblement aura dès lors la tâche d'autant plus difficile. La figure 3.16 présente la division de l'image du couple lorsqu'on a choisi une approximation par un polynôme de degré 3 et une valeur de T_1 à 5^2 , si l'on a une quantification à 8 bits du signal original. L'image qui en résulte est presque parfaite si ce n'est que le nombre de régions de la partition est très élevé, ce qui réduit considérablement la compression d'information.



a) Image approximée ($\alpha=3, T_1=5^2$)

b) Partition de l'image ($w=2020$).

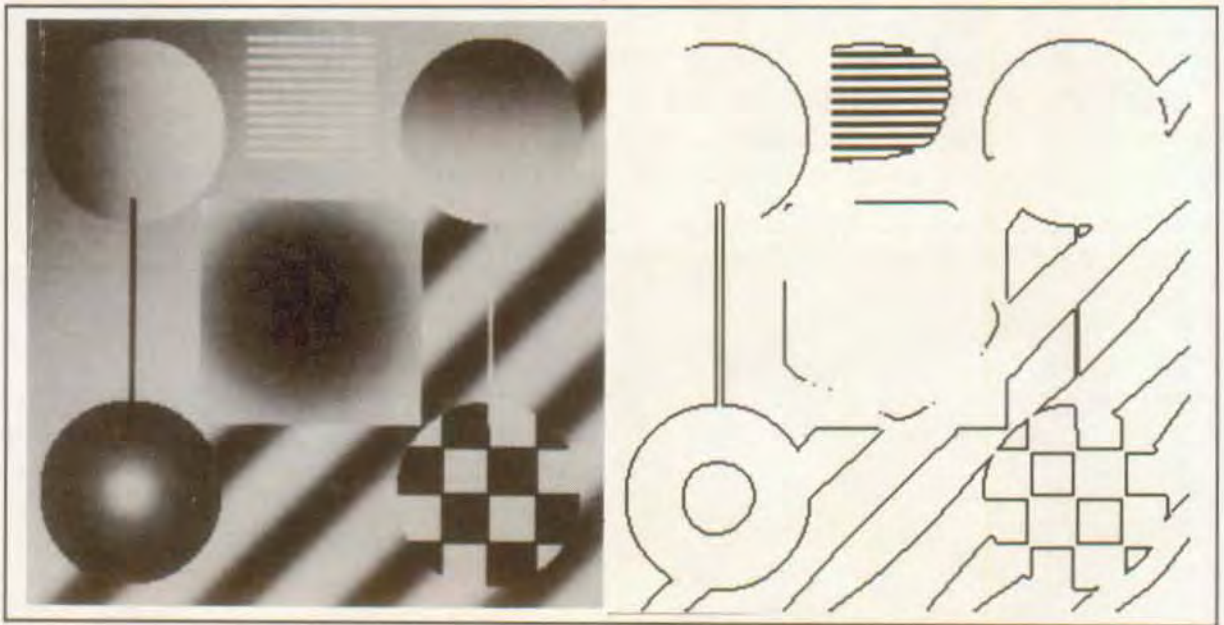
$$C_q = 976 \text{ bits} \quad C_l = 878 \text{ bits} \quad \rho = 3.23.$$

Figure 3.16: Division de l'image du couple sans image de contrôle des contours avec seuil sur l'erreur quadratique moyenne placé très bas.

De manière à éviter la perte de l'information contour sans toutefois subdiviser excessivement l'image, il suffit d'appliquer l'algorithme proposé à la figure 3.7 avec utilisation d'une image de contrôle des contours. Les résultats correspondants vont être présentés dans ce qui suit.

3.5.2 Effets des paramètres et compression:

Pour pouvoir estimer qualitativement les effets des paramètres de l'algorithme de division, ce dernier a été appliqué dans un premier temps à l'image de test de la figure 3.17a. La figure 3.17b définit l'image de contrôle des contours correspondante estimée au moyen du filtre haute fréquence de la section 2.2. Le tableau 3.18 indique les sous-numéros de la figure 3.19 pour les différentes valeurs du seuil et du degré du polynôme d'approximation dans l'application de l'algorithme de division à l'image de test.



a) Image de test (original)

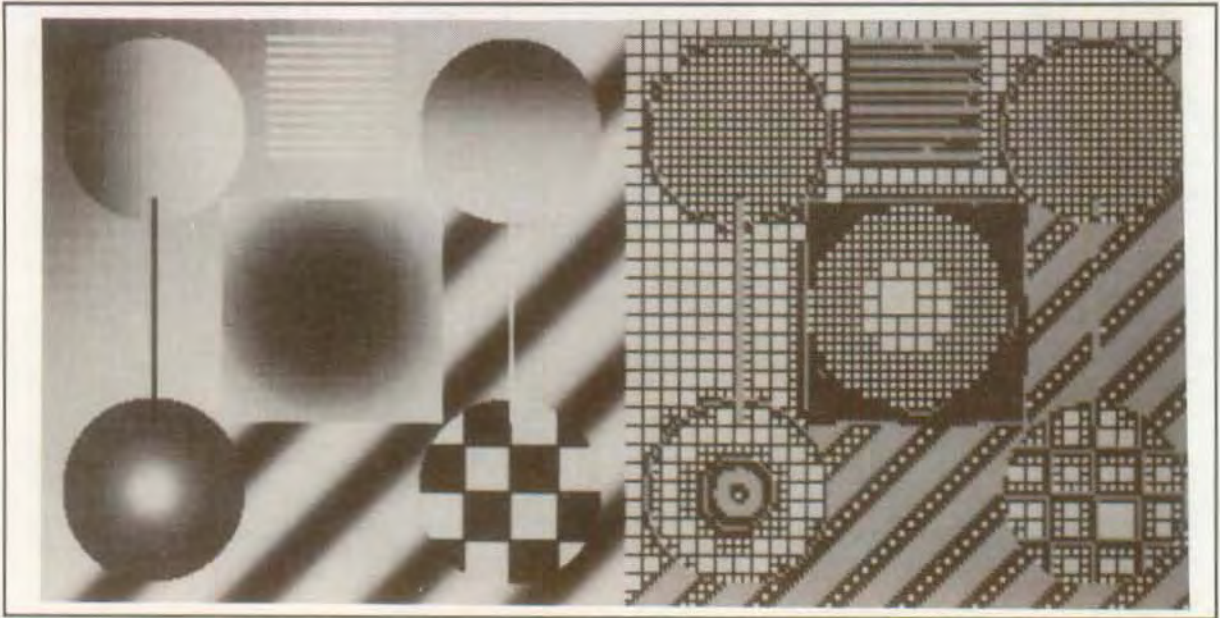
b) Image de contrôle des contours.

Figure 3.17: Image de test utilisée pour estimer l'effet des paramètres de l'algorithme de division de la figure 3.7.

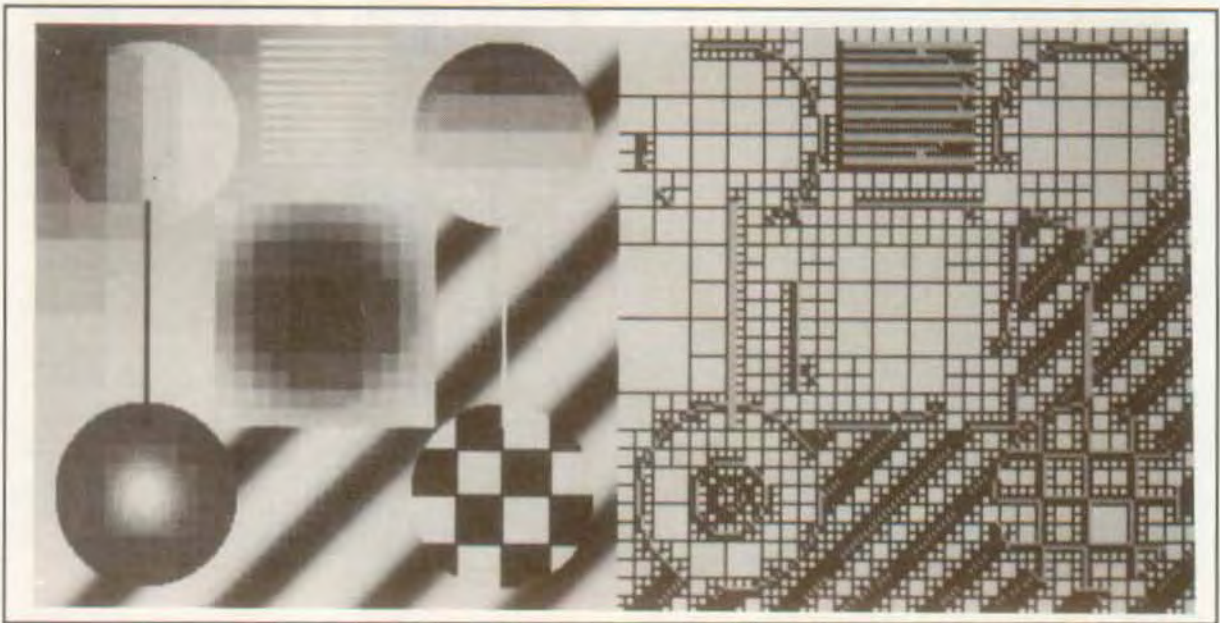
$\alpha \backslash T_1$	5^2	15^2	30^2
0	a	b	c
1	d	e	f
2	g	h	i
3	j	k	l

Tableau 3.18:

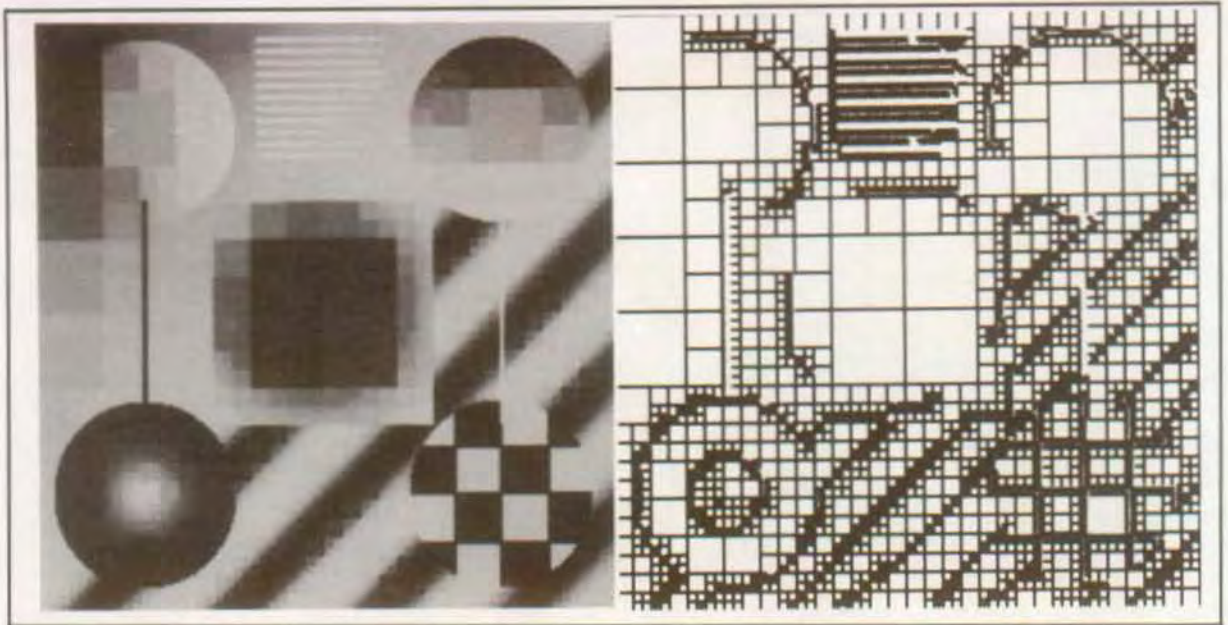
Numéros de la figure 3.19 associés à différentes valeurs des paramètres de seuil et de degré du polynôme.



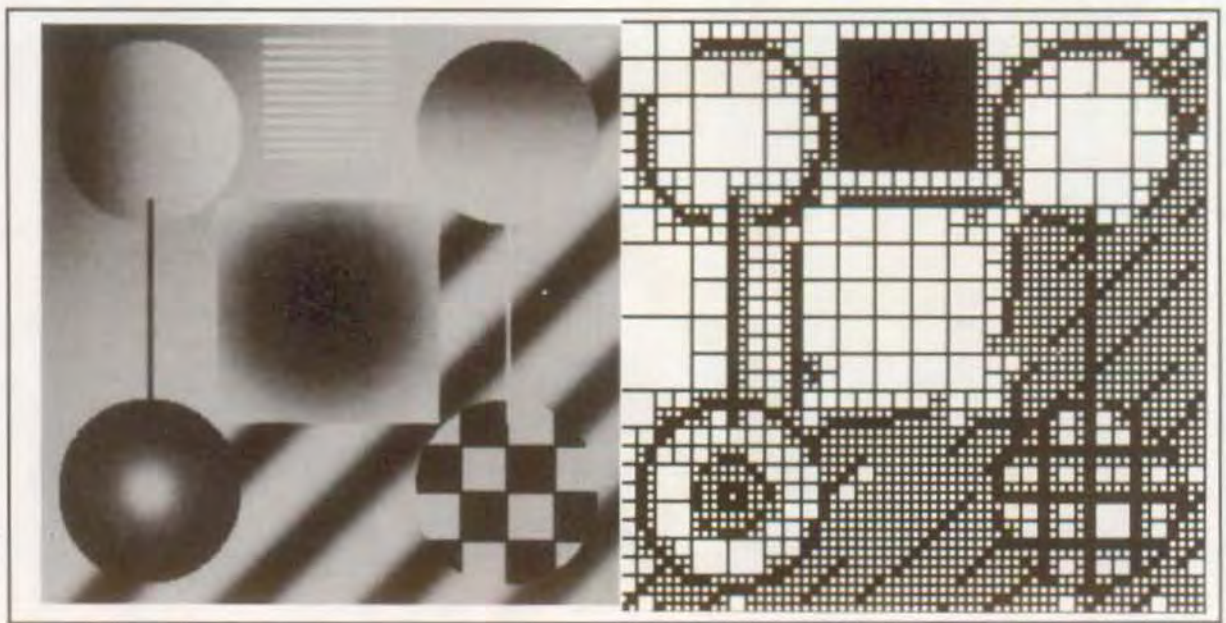
a) Image approximée ($\alpha=0, T_1=5^2$) et Partition de l'image ($w=21010$).
 $C_q = 11664$ bits $C_l = 11757$ bits $\rho = 2.92$.



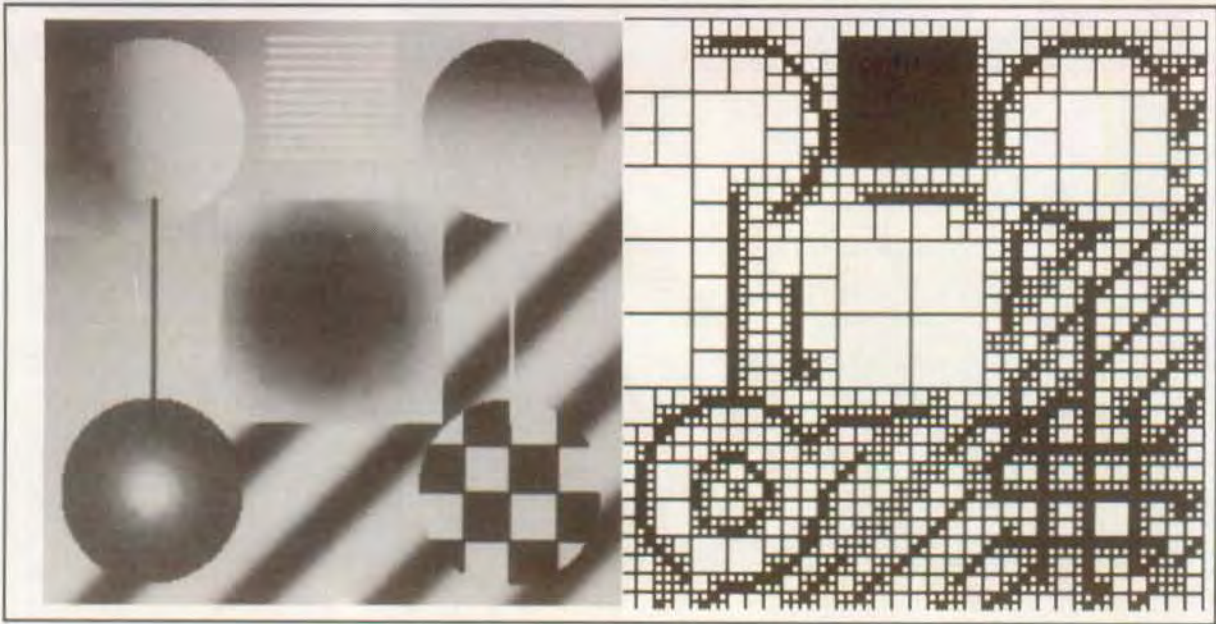
b) Image approximée ($\alpha=0, T_1=15^2$) et Partition de l'image ($w=9979$).
 $C_q = 7336$ bits $C_l = 8058$ bits $\rho = 6.01$.



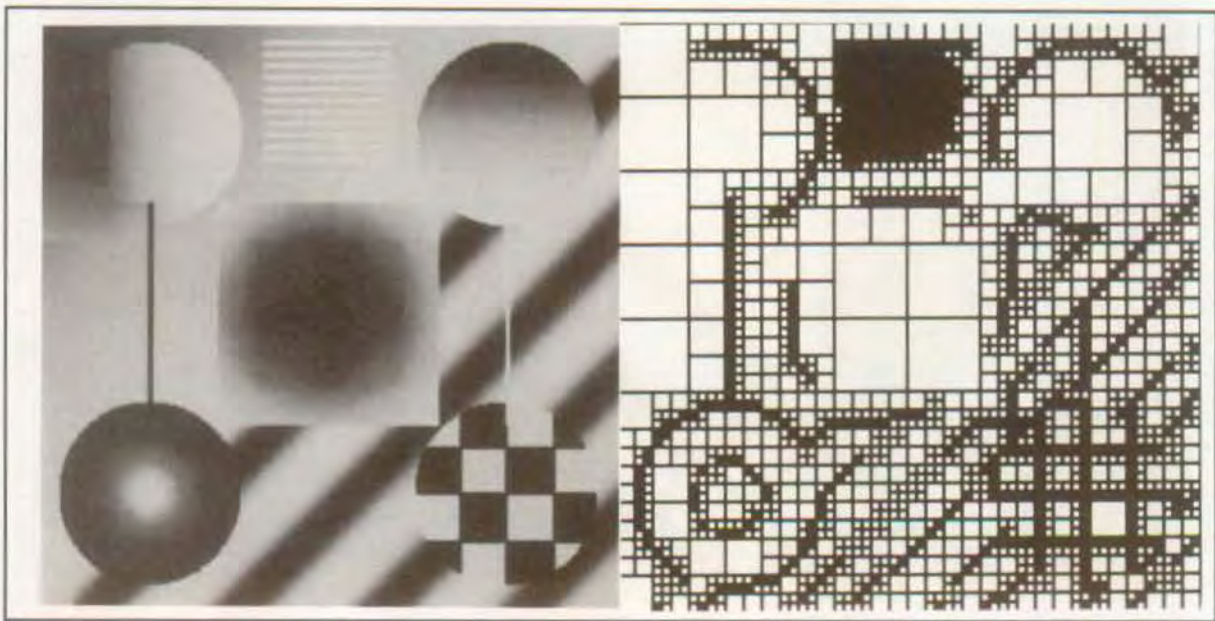
c) Image approximée ($\alpha=0, T_1=30^2$) et Partition de l'image ($w=8344$).
 $C_q = 5604$ bits $C_l = 6481$ bits $\rho = 7.25$.



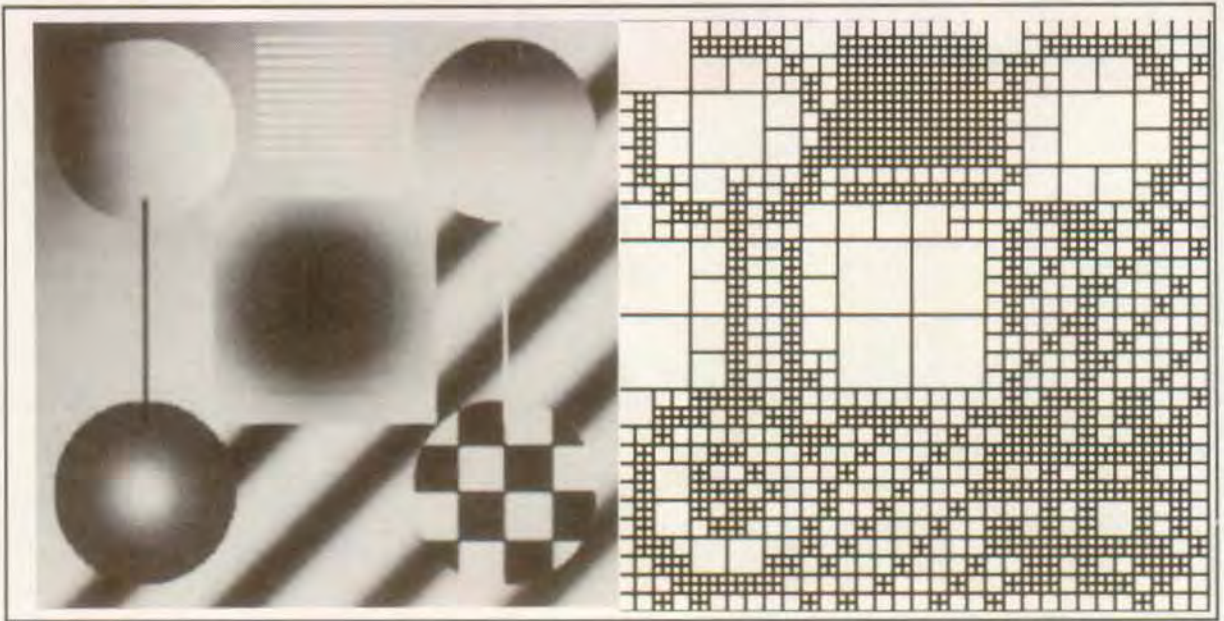
d) Image approximée ($\alpha=1, T_1=5^2$) et Partition de l'image ($w=4711$).
 $C_q = 3208$ bits $C_l = 3170$ bits $\rho = 4.51$.



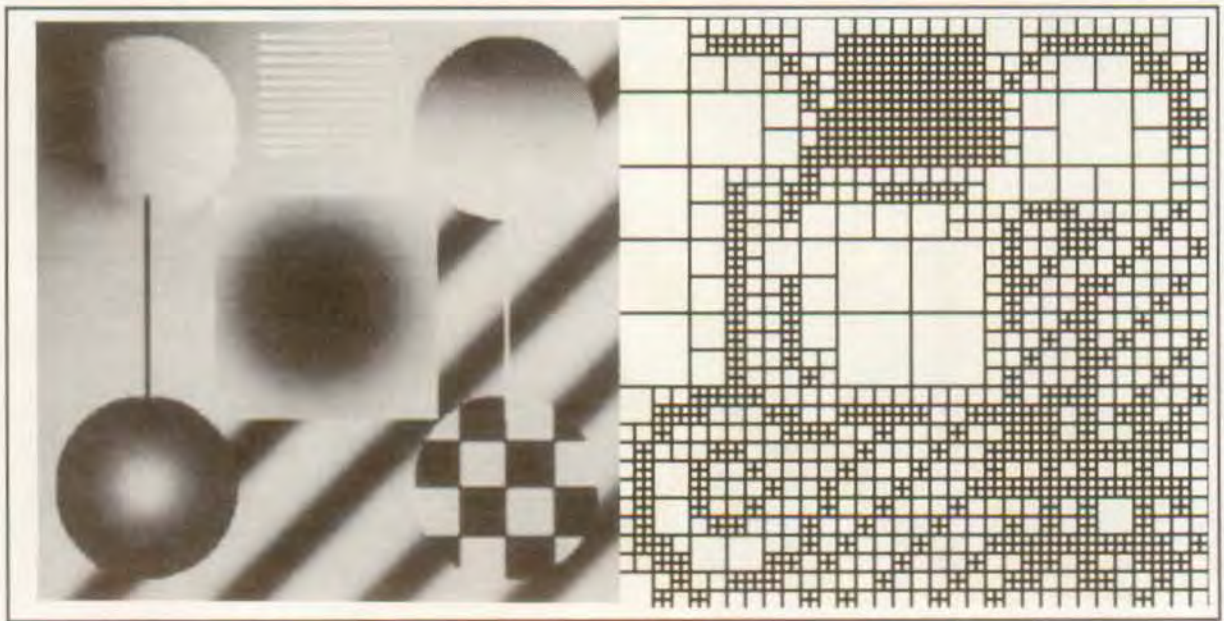
e) Image approximée ($\alpha=1, T_1=15^2$) et Partition de l'image ($w=3898$).
 $C_q = 2492$ bits $C_l = 2628$ bits $\rho = 5.46$.



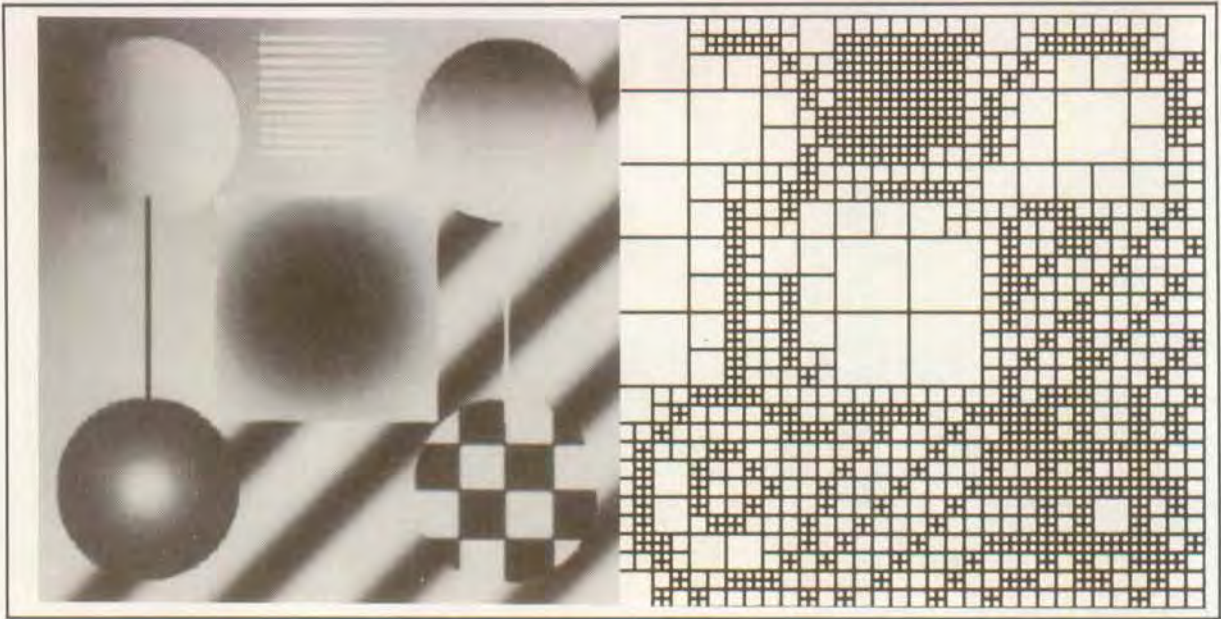
f) Image approximée ($\alpha=1, T_1=30^2$) et Partition de l'image ($w=3679$).
 $C_q = 2432$ bits $C_l = 2586$ bits $\rho = 5.78$.



g) Image approximée ($\alpha=2, T_1=5^2$) et Partition de l'image ($w=1813$).
 $C_q = 1036$ bits $C_l = 951$ bits $\rho = 5.90$.

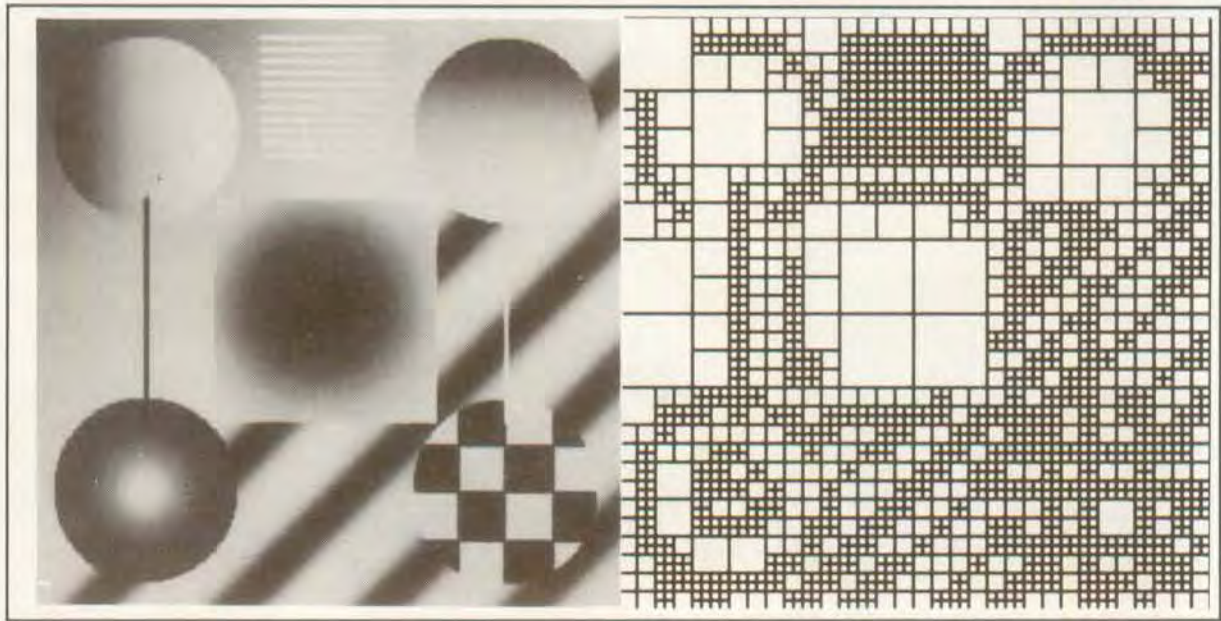


h) Image approximée ($\alpha=2, T_1=15^2$) et Partition de l'image ($w=1654$).
 $C_q = 984$ bits $C_l = 918$ bits $\rho = 6.53$.



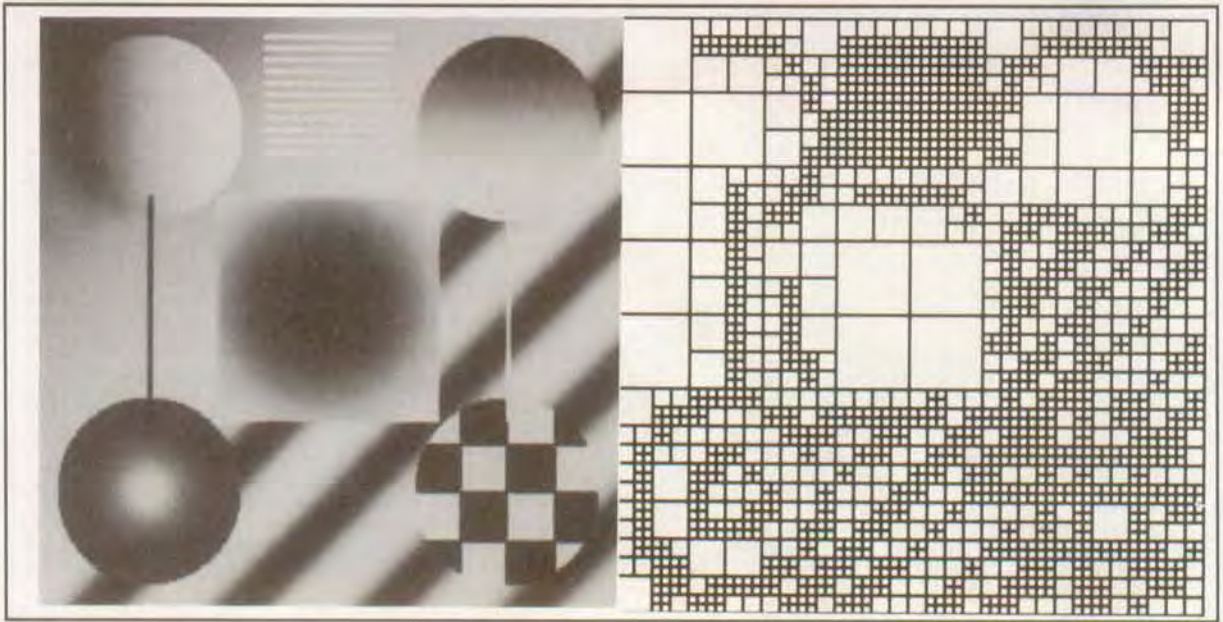
i) Image approximée ($\alpha=2, T_1=30^2$) et Partition de l'image ($w=1594$).

$$C_q = 984 \text{ bits} \quad C_l = 918 \text{ bits} \quad \rho = 6.77.$$



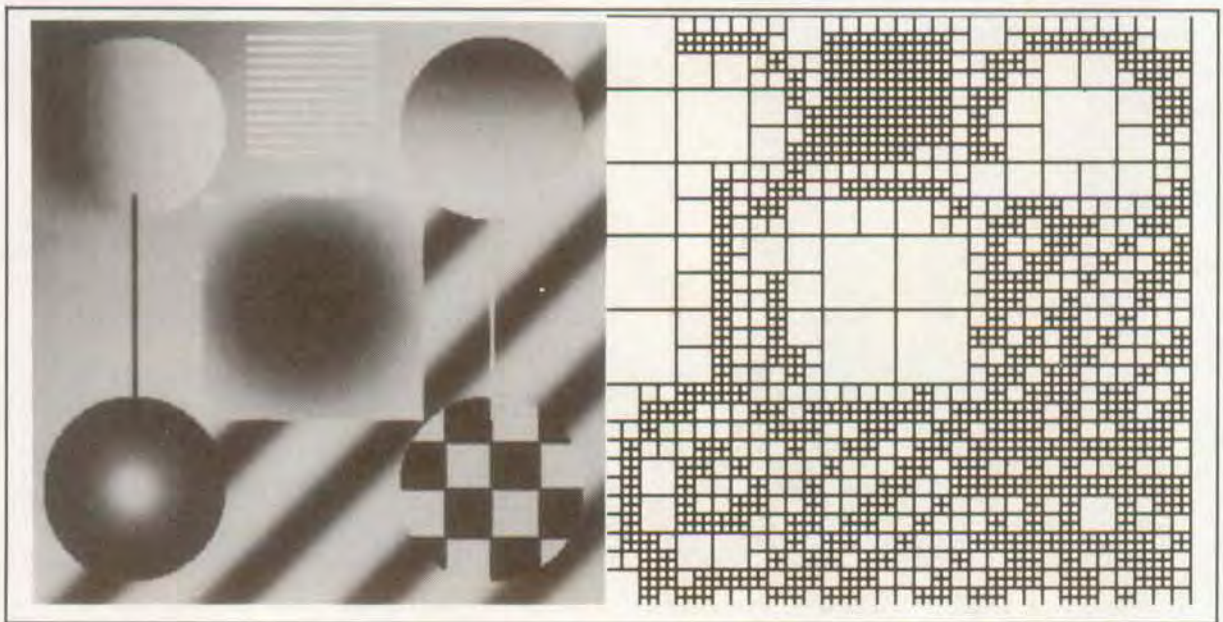
j) Image approximée ($\alpha=3, T_1=5^2$) et Partition de l'image ($w=1954$).

$$C_q = 1032 \text{ bits} \quad C_l = 945 \text{ bits} \quad \rho = 3.33.$$



k) Image approximée ($\alpha=3, T_1=15^2$) et Partition de l'image ($w=1849$).

$$C_q = 984 \text{ bits} \quad C_l = 911 \text{ bits} \quad \rho = 3.52.$$



l) Image approximée ($\alpha=3, T_1=30^2$) et Partition de l'image ($w=1819$).

$$C_q = 984 \text{ bits} \quad C_l = 911 \text{ bits} \quad \rho = 3.58.$$

Figure 3.19: Division de l'image de test avec différentes valeurs des paramètres.

Si l'image de contrôle des contours reproduit presque tous les contours, dans l'image reconstruite, ils restent préservés indépendamment de la valeur choisie pour T_1 .

Par contre, le degré du polynôme aura de l'influence sur la création de contours artificiels, si T_1 est trop élevé. En effet, plus la taille du carré est grande par rapport au nombre r des coefficients du polynôme, moins il est facile à la fonction d'approximation de suivre les variations du signal original au voisinage des frontières du carré. Bien entendu, ceci n'est pas valable si l'on se trouve dans un carré où la modification des paramètres u_i selon le critère des moindres carrés entraîne presque une interpolation des données de départ. Ce défaut peut être éliminé par l'application de l'algorithme de rehaussement décrit précédemment.

Enfin, on voit apparaître certains artefacts pour les carrés de plus petite taille contenant des contours importants si $r < 2 \times 2^l$, une interpolation des données de départ étant impossible. Les variations rapides entraînent des fluctuations incontrôlées de l'approximation polynomiale. En garantissant une correspondance entre frontières de régions et contours réels de l'image, la segmentation finale permettra d'éliminer ce genre d'erreurs.

Pour ce qui est de la forme de la segmentation, le nombre de carrés décroît avec l'augmentation du degré de l'approximation. La différence provient essentiellement des carrés de plus petite taille. En effet, la plupart des carrés apparaissent au voisinage des contours de l'image, à cause du critère de division choisi. Ceux-ci sont toujours de la plus petite taille possible (1x1 pour le degré 0, 2x2 pour le degré 1, 4x4 pour les degrés 2 et 3). Il existe donc pratiquement 4 et 16 fois plus de carrés de plus petite taille pour le polynôme de degré 1 et 0 que pour ceux de degrés 1 et 2, respectivement. Si reconstruire la partition est relativement plus coûteux sur le plan de la compression pour les polynômes de degré faible, il n'en va pas de même pour la mémorisation des coefficients d'approximation, leur nombre augmentant avec le degré du polynôme. On peut choisir de mémoriser seulement un certain nombre de coefficients, encore faut-il indiquer lesquels on a choisi et selon quel critère.

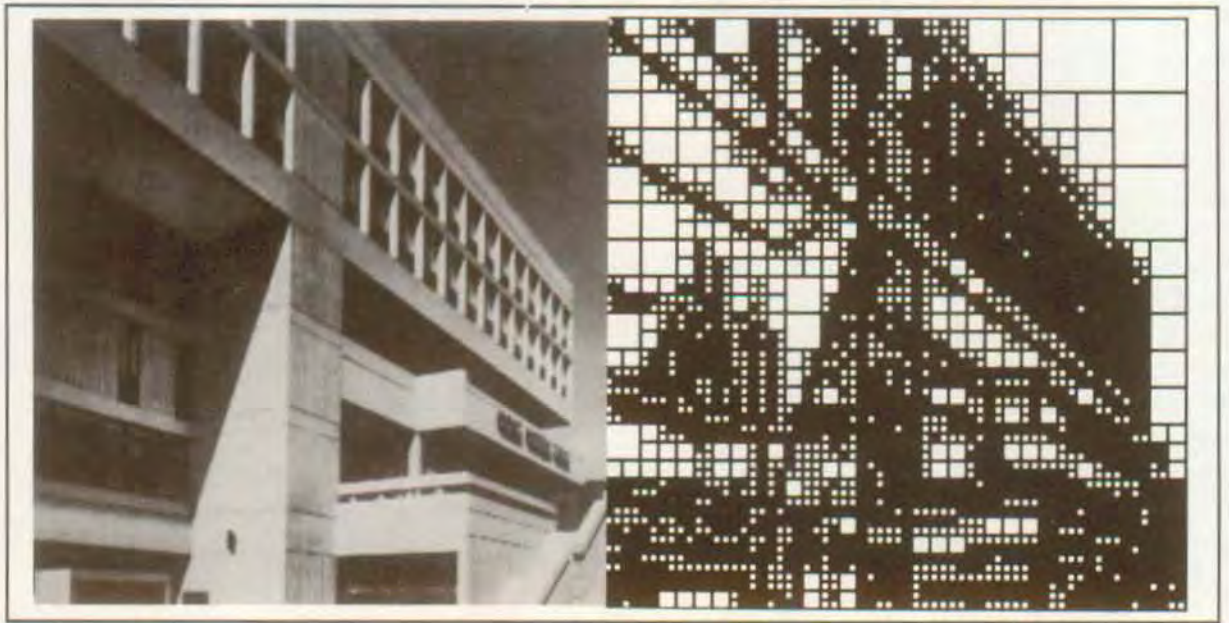
Ces différentes interprétations seront confirmées par des exemples sur les images naturelles des figures 2.2 à 2.4.

Rôle du seuil sur l'erreur quadratique moyenne:

De ce qui précède, il résulte que les contours existants de l'image sont préservés grâce à l'utilisation de l'image de contrôle. Le seuil T_1 permet de limiter le nombre de carrés ne présentant pas de contours tout en garantissant une bonne correspondance moyenne entre le polynôme approximant et le signal à l'intérieur de chaque carré. Ceci permettra d'accélérer le fonctionnement de l'algorithme de rassemblement, un moins grand nombre de régions devant être prises en considération. Néanmoins, il ne faut pas placer ce seuil à un niveau trop élevé, sans quoi on risquerait de ne pas avoir la meilleure correspondance entre fonctions polynomiales et les données originales après segmentation. En effet, lors de la procédure adaptative de rassemblement, on cherche toujours le lieu de dissimilarité minimum du graphe de contiguïté de régions pour atteindre un

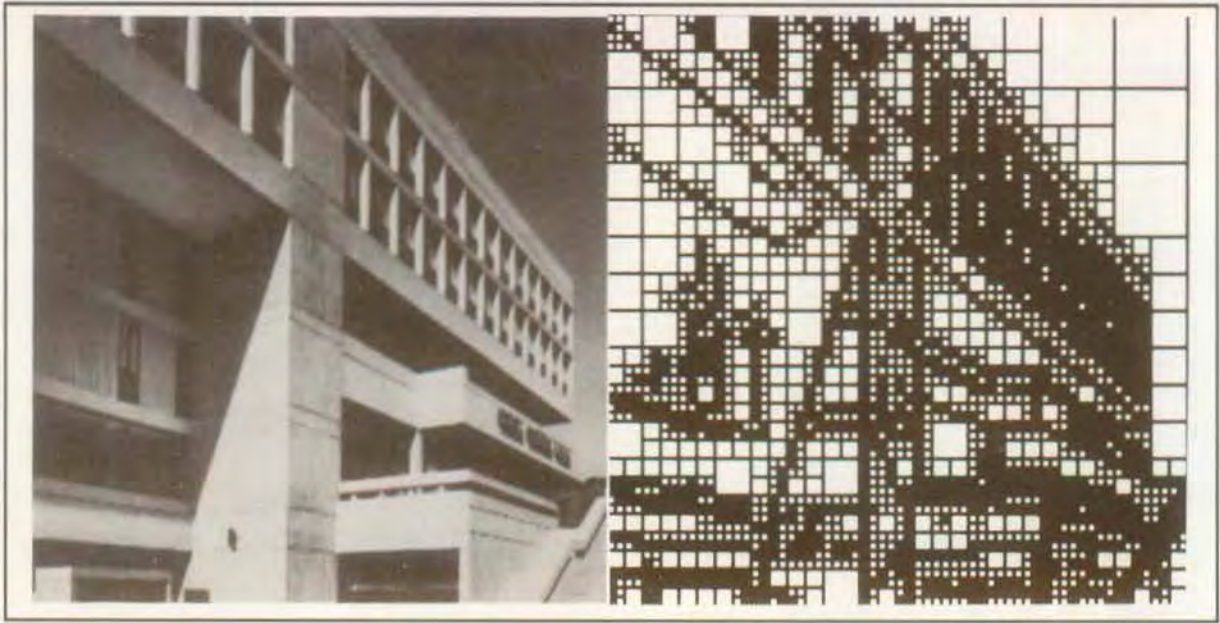
optimum. Celui-ci serait faussé par la structure initiale du graphe imposée par l'algorithme de division.

La figure 3.20 présente les résultats de l'algorithme de division appliqué au bâtiment pour trois valeurs de seuil (5^2 , 15^2 , 30^2) et une fonction polynômiale séparable de degré 2 définie par la relation (3.9) avec $r_1=r_2=2$. La figure 3.21 illustre la partition et l'approximation que l'on obtient de l'image du caméraman après application de l'algorithme de division en utilisant des polynômes de degré 2 et les mêmes valeurs pour le seuil T_1 que dans la figure précédente. En observant les figures 3.20 et 3.21, on remarque que la qualité des résultats reste inchangée quelle que soit la valeur de T_1 . En effet, à ce niveau de la segmentation les polynômes sont parfaitement à même de suivre des régions homogènes à variation lente. En comparant les deux figures, on ne remarque pas de dégradation au voisinage des contours de la figure 3.20, le polynôme choisi interpolant parfaitement les données images pour tous les carrés de taille 2×2 . Dans le cas de la figure 3.21, le polynôme est en mesure de suivre les variations brusques du signal luminance à l'emplacement des contours, mais des effets de bord sont visibles. En effet, dans ce cas, l'approximation au sens des moindres carrés ne peut interpoler les pixels du carré 4×4 (6 coefficients d'approximation contre 16 pixels).

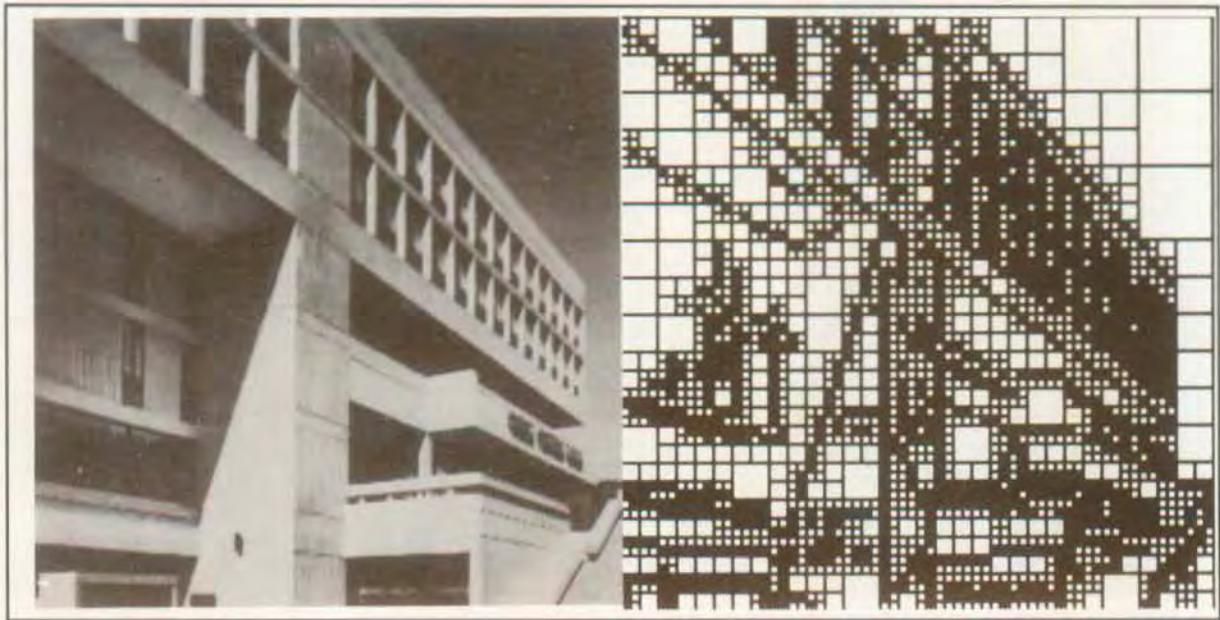


a) Image approximée ($r_1=2, r_2=2, T_1=5^2$) et Partition de l'image ($w=8749$).

$$C_q = 3996 \text{ bits} \quad C_l = 3477 \text{ bits} \quad \rho = 1.85.$$

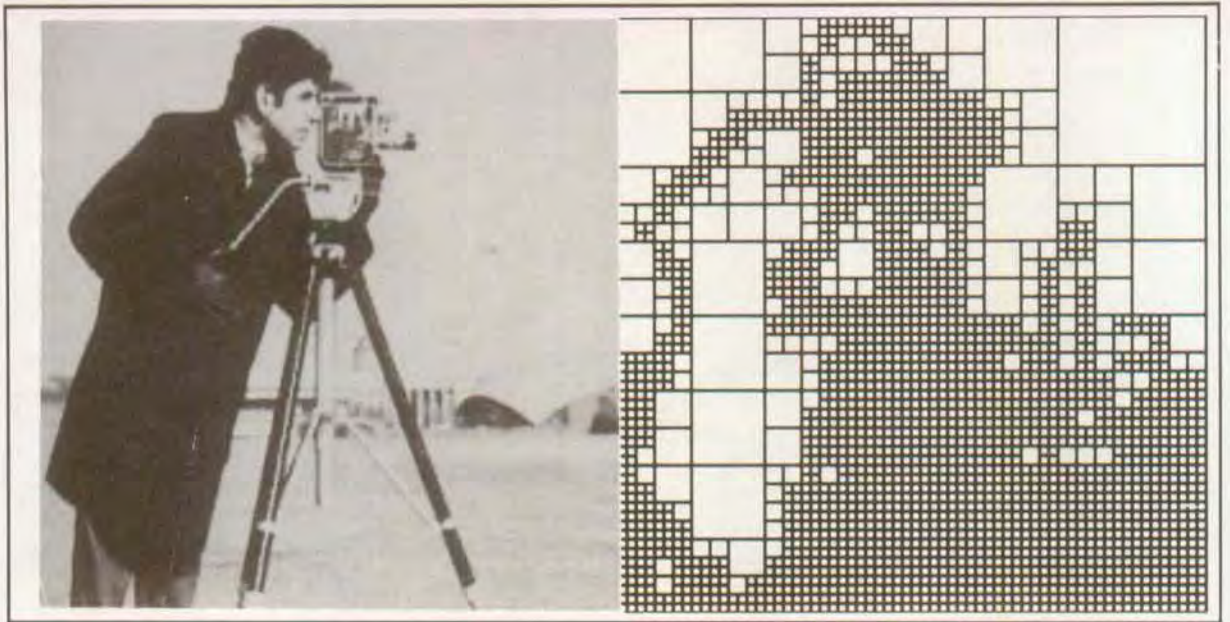


b) Image approximée ($r_1=2, r_2=2, T_1=15^2$) et Partition de l'image ($w=6847$).
 $C_q = 3608$ bits $C_1 = 3302$ bits $\rho = 2.36$.

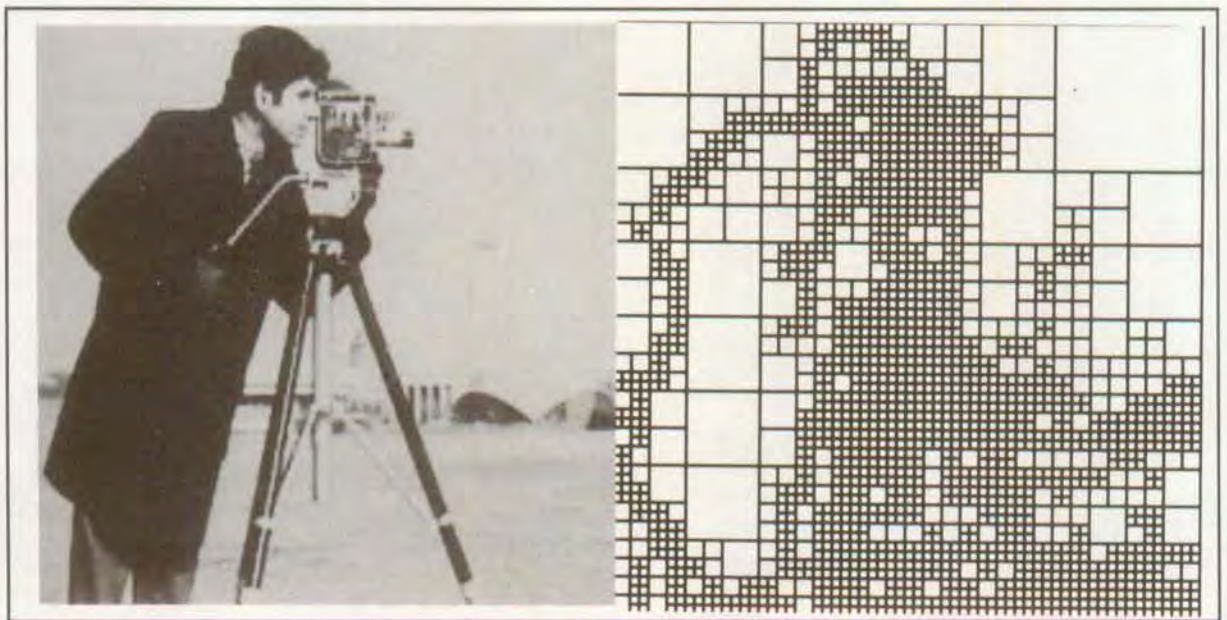


c) Image approximée ($r_1=2, r_2=2, T_1=30^2$) et Partition de l'image ($w=6391$).
 $C_q = 3544$ bits $C_1 = 3294$ bits $\rho = 2.52$.

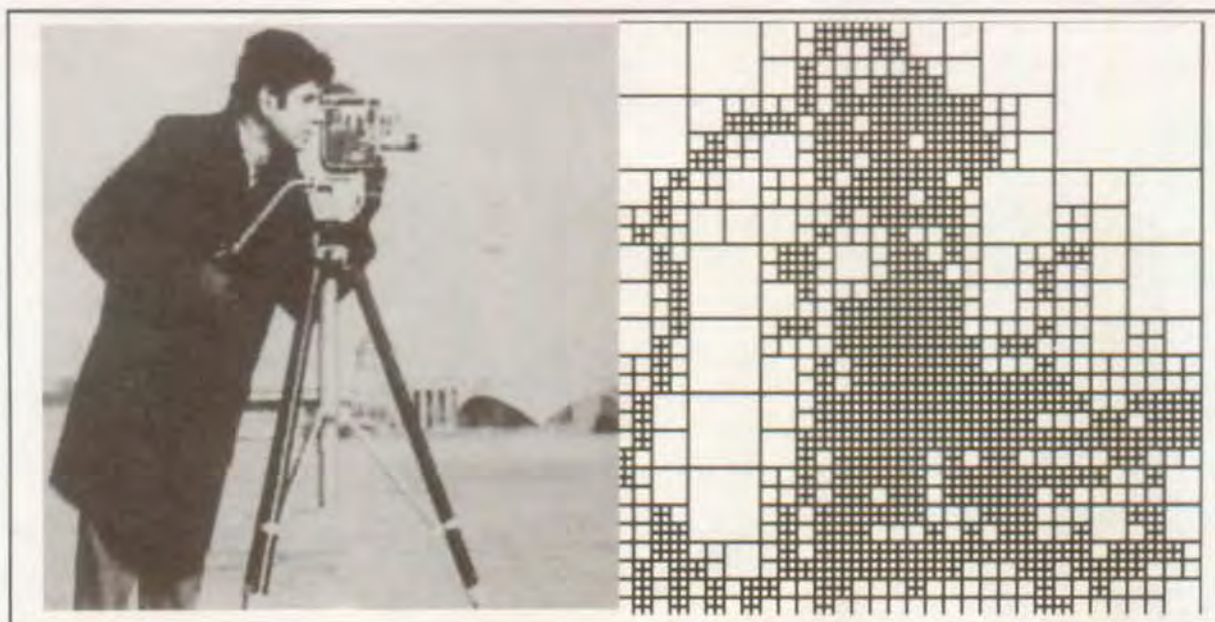
Figure 3.20: Division de l'image du bâtiment pour trois valeurs différentes de seuil sur l'erreur quadratique moyenne et une fonction polynomiale séparable.



a) Image approximée ($\alpha=2, T_1=5^2$) et Partition de l'image ($w=2293$).
 $C_q = 924$ bits $C_l = 799$ bits $\rho = 4.73$.



b) Image approximée ($\alpha=2, T_1=15^2$) et Partition de l'image ($w=1924$).
 $C_q = 904$ bits $C_l = 813$ bits $\rho = 5.63$.



c) Image approximée ($\alpha=2, T_1=30^2$) et Partition de l'image ($w=1714$).
 $C_q = 900$ bits $C_l = 822$ bits $\rho = 6.31$.

Figure 3.21: Division de l'image du caméraman pour trois valeurs différentes de seuil sur l'erreur quadratique moyenne et une fonction polynomiale de degré 2.

Dans ce qui suit, on a décidé de fixer systématiquement T_1 à 15^2 , indépendamment de l'image traitée.

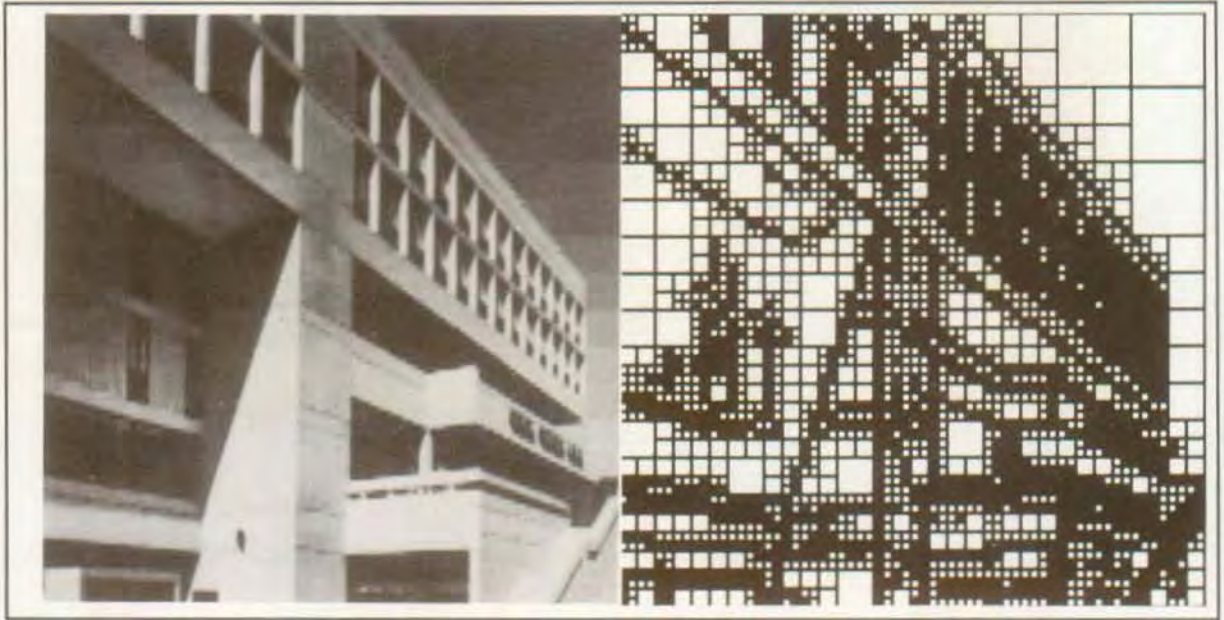
Rôle du degré du polynôme:

Si l'on n'utilise pas d'image de contrôle des contours, l'augmentation du degré du polynôme permet de suivre dans une certaine mesure les sauts de luminance (voir les images des figures 3.13c et 3.14b).

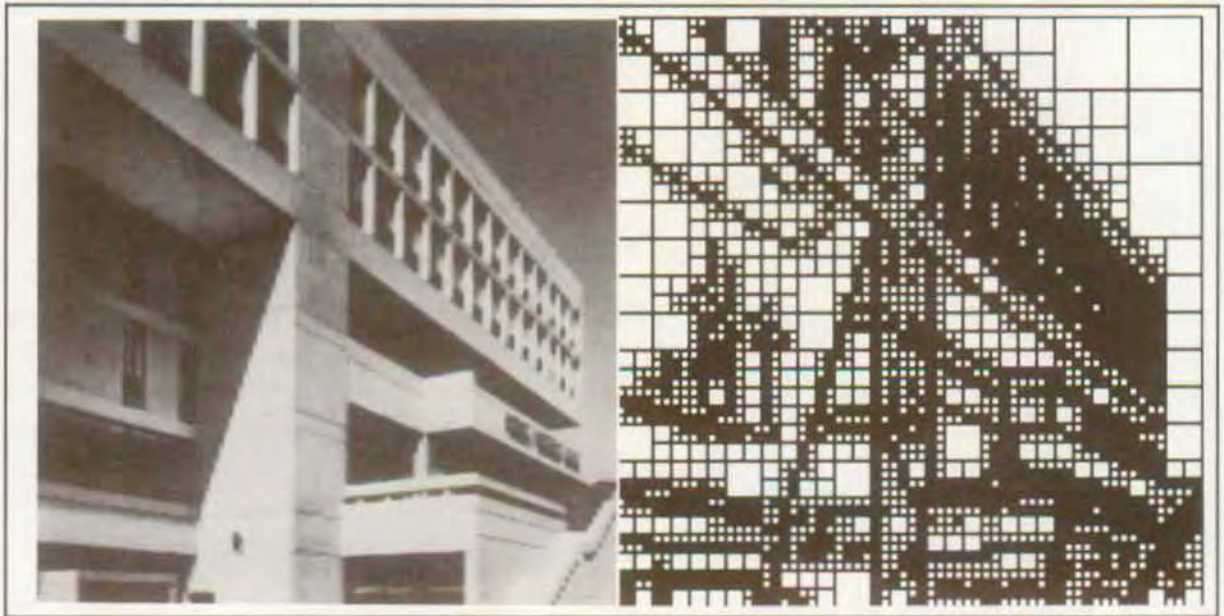
Comme l'algorithme utilisé est celui de la figure 3.7, il paraît plus judicieux de baser le choix du degré du polynôme en liaison avec le rapport de compression. En gardant le même seuil pour T_1 qui permet d'estimer localement la qualité d'un carré ne contenant pas de contour, on choisit le degré du polynôme qui permet d'aboutir à une partition de l'image correspondant à un coût minimal en nombre de bits. Ce coût est toutefois difficile à estimer sans quantifier d'une façon vectorielle les coefficients d'approximation.

Comme ce sujet n'a pas été abordé au niveau de l'algorithme de division, on peut remarquer que le degré du polynôme d'approximation doit être choisi de façon à reproduire la plupart des caractéristiques de la scène. Le problème reste que l'algorithme ne peut savoir à l'avance quelles sont les caractéristiques de luminance de chaque région. Plus on augmente le degré du polynôme, plus le nombre de surfaces que l'on peut décrire augmente. Ceci se fait cependant au détriment de la compression puisque chaque carré sera caractérisé par un nombre plus élevé de coefficients non absolument nécessaires. En utilisant successivement des polynômes séparables de degrés respectifs 0 et 1, d'une part, 1 et 0, d'autre

1 et 0, d'autre part, la figure 3.22 indique les défauts introduits au niveau de l'image du bâtiment. Cette figure est à comparer à la figure 3.20. L'augmentation du degré selon x ou y permet de garder dans une certaine mesure la continuité de la luminance entre les différents carrés de grande taille.



a) Image approximée ($r_1=1, r_2=2, T_1=15^2$) et Partition de l'image ($w=7111$).
 $C_q = 3640$ bits $C_l = 3311$ bits $\rho = 4.48$.



b) Image approximée ($r_1=2, r_2=1, T_1=15^2$) et Partition de l'image ($w=7087$).
 $C_q = 3636$ bits $C_l = 3304$ bits $\rho = 4.49$.

Figure 3.22: Division de l'image du bâtiments pour des fonctions polynomiales séparables de degré 1 selon un axe et 0 selon l'autre.

En observant ces figures, on remarque que la qualité n'est pas parfaite au niveau des carrés de plus petite taille (2x2) contenant des contours car le nombre de coefficients (2) est inférieur au nombre de points contenu dans chacun de ces carrés.

Aspects liés à la compression de l'information:

Ce paragraphe discute de la compression qui est réalisée à ce niveau de la segmentation. En ce qui concerne le codage de position et de taille des carrés, on remarque que le codage conditionnel par taille de carrés est souvent plus efficace que le codage par arbre quaternaire, d'environ 10%. Ceci confirme les hypothèses faites au paragraphe 3.4.1.

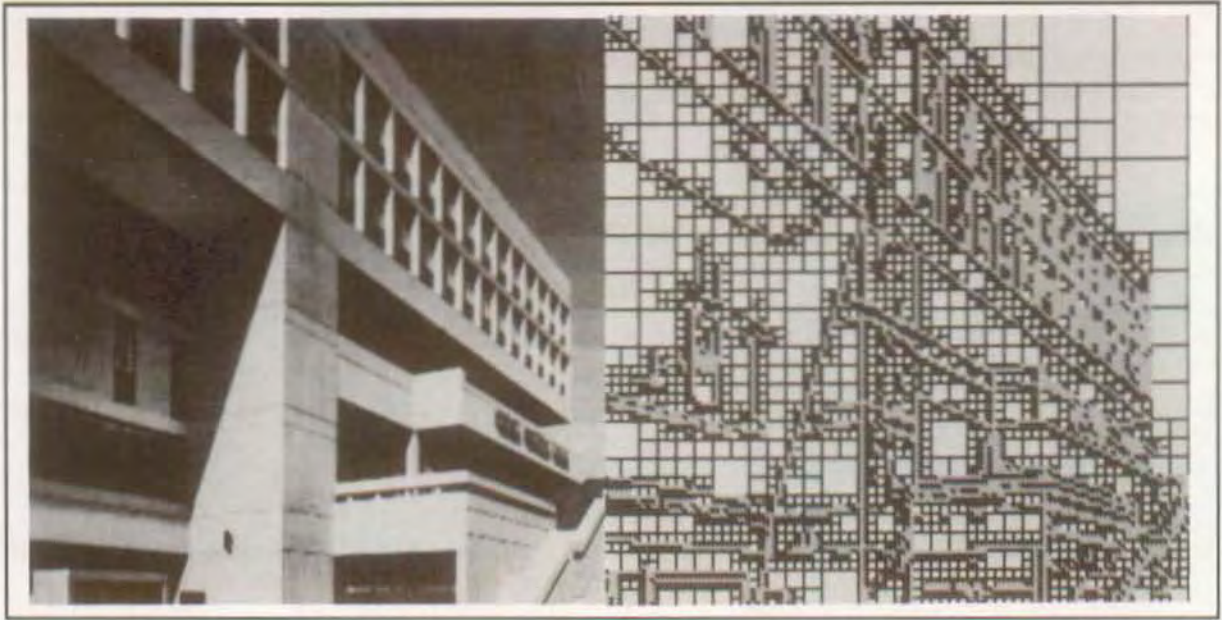
La méthode de codage des coefficients polynomiaux n'est pas optimale ainsi qu'il a été souligné à la section 3.4. Les images présentées aux paragraphes précédents indiquent à chaque fois le rapport de compression ρ .

Comme dans le codage des coefficients polynomiaux, on ne tient absolument pas compte de la corrélation qui existe entre les différents carrés, on présente dans ce qui suit les résultats obtenus par le CDC en approximant chaque carré par sa valeur moyenne (polynôme de degré 0). L'espace des différences est quantifié à 16 niveaux. Après codage de Huffman de ces 16 symboles, le rapport de compression s'exprime par la relation:

$$\rho = 256^2 \times 8 / (C_c + \min(C_q, C_l)) = 524288 / (C_c + \min(C_q, C_l)) \quad (3.20)$$

où C_c représente le coût lié à la transmission des valeurs moyennes de chaque carré.

Les figures 3.23 à 3.25 montrent la reconstitution que l'on obtient en adoptant cette stratégie pour les images du bâtiment, du caméraman et du couple. Chaque figure présente successivement la partition de l'image en carrés (a), sa reconstruction sans quantification de l'espace des différences (b) et l'image codée et restaurée (c).

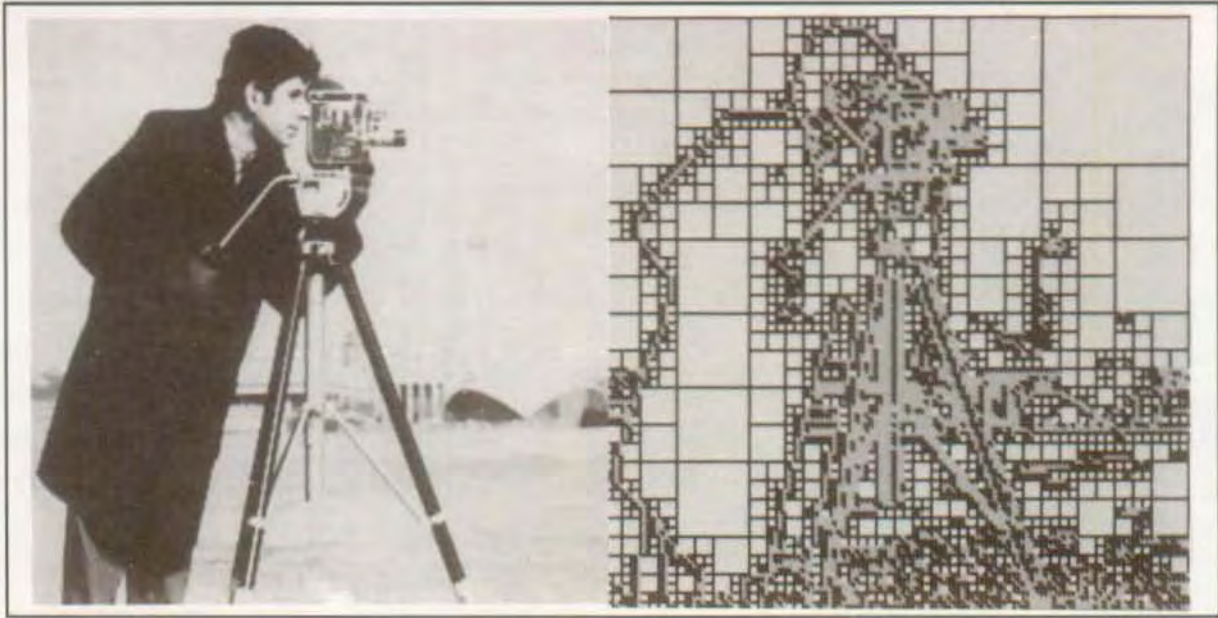
a) Image approximée ($\alpha=0, T_1=15^2$)b) Partition de l'image ($w=17635$).

$C_q = 10016$ bits $C_l = 9933$ bits $\rho = 12.62$.



c) Image codée et rehaussée.

Figure 3.23: Division de l'image du bâtiment avec une fonction d'approximation par niveaux constants. Image codée et rehaussée par CDC 16 bits.

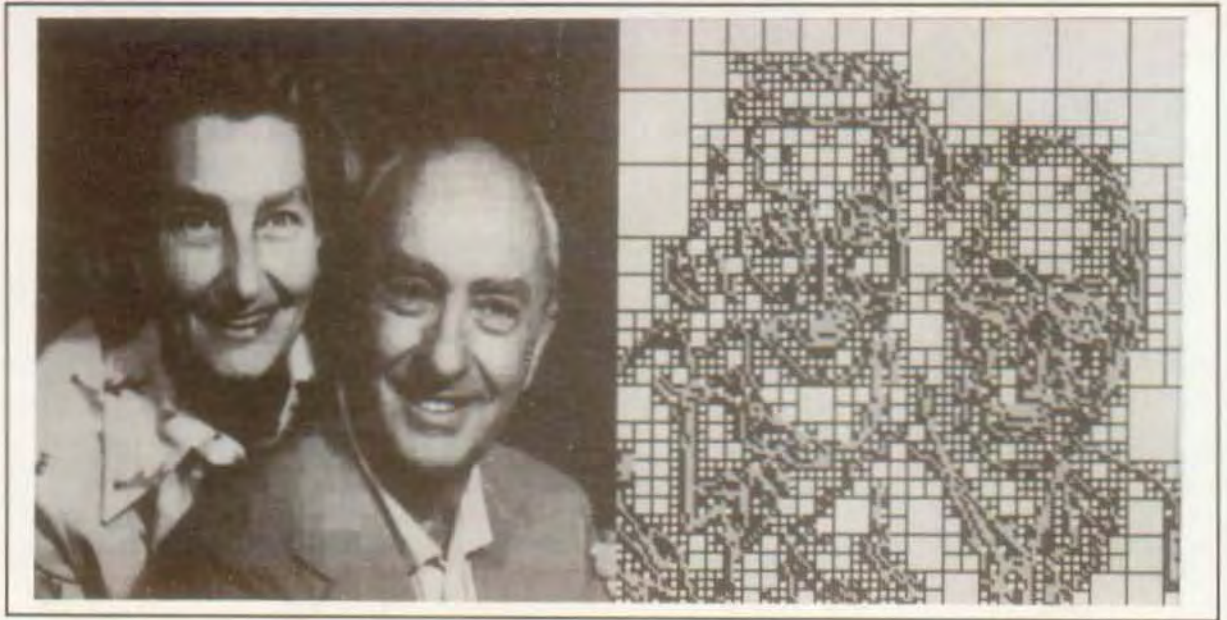
a) Image approximée ($\alpha=0, T_1=15^2$)b) Partition de l'image ($w=14239$).

$C_q = 7844$ bits $C_l = 7765$ bits $\rho = 15.58$.



c) Image codée et rehaussée.

Figure 3.24: Division de l'image du cameraman avec une fonction d'approximation par niveaux constants. Image codée et rehaussée par CDC 16 bits.

a) Image approximée ($\alpha=0, T_1=15^2$)b) Partition de l'image ($w=13903$).

$$C_q = 8756 \text{ bits} \quad C_l = 8952 \text{ bits} \quad \rho = 17.96.$$



c) Image codée et rehaussée.

Figure 3.25: Division de l'image du couple avec une fonction d'approximation par niveaux constants. Image codée et rehaussée par CDC 16 bits.

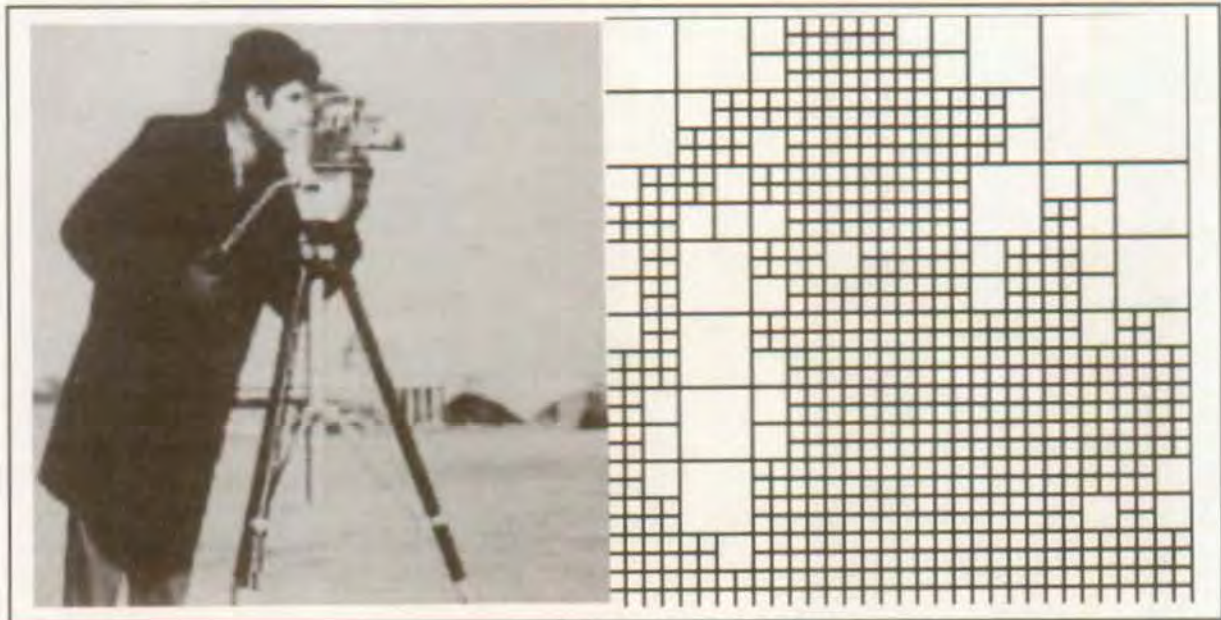
Il est important de remarquer que le codage de position permettant de reconstituer la structure de la segmentation après division représente une part négligeable du coût associé à la reconstitution de l'image approximée. Certes, la quantification des coefficients n'a pas été étudiée dans les détails pour réduire au mieux la corrélation qui existe entre l'ensemble de coefficients associé à chaque carré. Il n'en reste pas moins que la structure simple de la segmentation après division rend son codage peu coûteux. Après le processus de rassemblement, la

situation sera inversée. Le codage des coefficients prendra une part négligeable par rapport au codage de forme et de position de chaque région.

3.5.3 Limitation de la plus petite taille de carré:

Lorsque le nombre de coefficients du polynôme n'est pas beaucoup plus petit que le nombre de points à l'intérieur d'un carré, on a vu que l'approximation pouvait suivre dans une certaine mesure le signal original même lorsque le carré considéré contient un contour. Si on décide de limiter la plus petite taille de carré, on risque d'avoir une représentation relativement fidèle du signal original tout en diminuant sensiblement le nombre de carrés à coder.

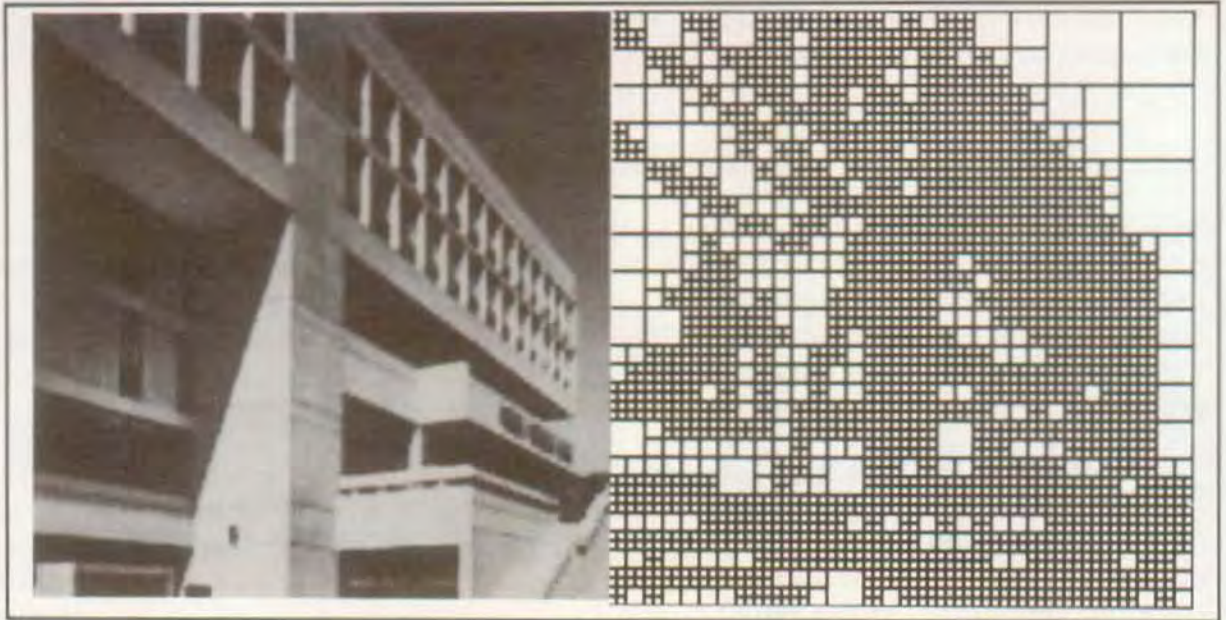
A cet effet, la figure 3.26 présente l'image du caméraman approximée par des polynômes de degré 3 où la plus petite taille de carré a été limitée à 8×8 . De la même manière, on présente à la figure 3.28 l'image du bâtiment approximée par des plans lorsqu'on a limité la plus petite taille de carré à 4×4 . Enfin, la figure 3.28 présente l'image du couple sachant que l'on a utilisé le même schéma de codage que pour la figure 3.25 alors que les carrés les plus petits sont de taille 2×2 .



a) Image approximée ($\alpha=3$ $T_1=15^2$) b) Partition de l'image ($w=682$).

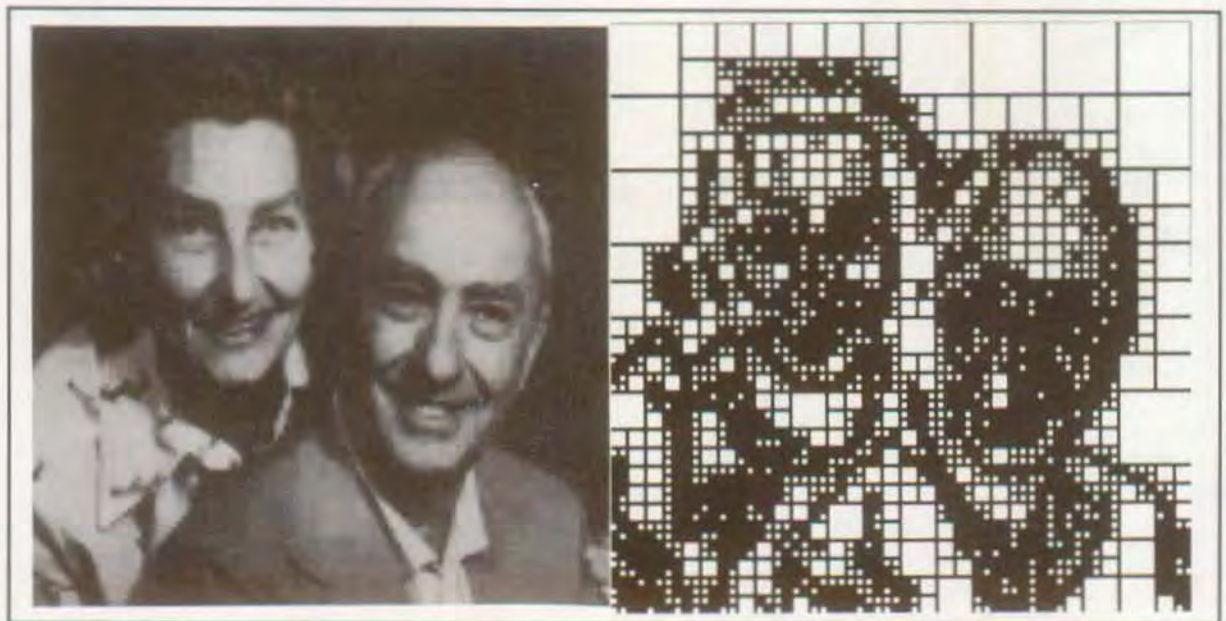
$$C_q = 272 \text{ bits} \quad C_l = 232 \text{ bits} \quad \rho = 9.57.$$

Figure 3.26: Division de l'image du caméraman avec une fonction d'approximation par polynômes de degré 3. La division est limitée à des carrés de taille 8×8 .



a) Image approximée ($\alpha=3$ $T_1=15^2$) b) Partition de l'image ($w=2713$).
 $C_q = 1140$ bits $C_l = 954$ bits $\rho = 7.94$.

Figure 3.27: Division de l'image du bâtiment avec une fonction d'approximation par polynômes de degré 1. La division est limitée à des carrés de taille 4×4 .



a) Image approximée ($\alpha=0, T_1=15^2$) b) Partition de l'image ($w=6571$).
 $C_q = 3256$ bits $C_l = 3001$ bits $\rho = 41.02$.



c) Image codée et rehaussée.

Figure 3.28: Division de l'image du couple avec une fonction d'approximation par niveaux constants. Image codée et rehaussée par CDC 16 bits. Plus petite taille de carrés = 2×2 .

3.6 Conclusion :

Dans ce chapitre, on a étudié la première phase de la segmentation : la division de l'image en un ensemble de carrés de différentes tailles, chacun d'eux correspondant à une partie d'espace régulier.

Dans un premier temps, le critère de division a été défini en termes généraux: un carré est divisé chaque fois que l'approximation polynomiale des données qu'il contient ne permet pas de reproduire l'information contour (l'erreur de contour est trop élevée), ou lorsque l'erreur quadratique moyenne dépasse un certain seuil.

Dans un deuxième temps, on a présenté des implantations possibles de l'algorithme suivies d'une analyse de la complexité des traitements à effectuer. De celle-ci, il ressort qu'il est possible de simplifier considérablement le problème en remplaçant la mesure sur l'erreur de contour par un simple test de présence d'un contour à l'intérieur d'un carré donné.

La section suivante traite de la représentation de l'information segmentée après application de l'algorithme de division. Deux codes sont définis pour décrire la structure de la segmentation: le premier utilise la notion d'arbre quaternaire; le second utilise la probabilité d'apparition de chaque taille de carré tout en conservant les contraintes géométriques imposées par l'algorithme de division. Lorsque certaines hypothèses sont valables, ce qui est souvent le cas (voir section 3.5), ce code s'avère plus efficace (10%) que le précédent. Le codage de l'approximation n'est pas traité en détail dans ce chapitre ce qui réduit les rapports de compression que l'on peut estimer après division. Un code simple a été cependant développé pour diminuer la redondance existant entre les

approximations des différents carrés dans le cas d'une approximation par constantes.

Enfin, la dernière section illustre expérimentalement les différents paramètres de l'algorithme. En particulier, la forme de la segmentation apparaît plus régie par les conditions imposées par l'image de contrôle des contours que par le seuil placé sur la mesure de l'erreur quadratique moyenne. L'augmentation du degré du polynôme ne fait que limiter la plus petite taille de carré sans changer la partition de l'image pour les régions de plus grande taille. La texture correspondant aux carrés de grande taille est cependant plus fidèlement reproduite en augmentant le degré du polynôme; pour les carrés de plus petite taille contenant des contours, on peut voir apparaître certains effets de blocs lorsque le nombre de coefficients d'approximation reste trop inférieur au nombre de points du carrés. Cette section donne aussi à titre indicatif les rapports de compression qui peuvent être obtenus à ce niveau. En utilisant une approximation par constantes, une reconstitution acceptable de l'image du couple a pu être obtenue avec un rapport de compression dépassant 40 à 1. Il nous reste à voir comment ceci peut être amélioré en réduisant encore la redondance d'information par le processus de rassemblement.