HAL
open science

Mise en oeuvre d'un système adaptatif de segmentation d'images

Christophe Rosenberger

To cite this version:

Christophe Rosenberger. Mise en oeuvre d'un système adaptatif de segmentation d'images. Interface homme-machine [cs.HC]. Université Rennes 1, 1999. Français. NNT : . tel-00194453

HAL Id: tel-00194453

https://theses.hal.science/tel-00194453

Submitted on 6 Dec 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N^o d'ordre : 2292

THÈSE

présentée

DEVANT L'UNIVERSITÉ DE RENNES 1

pour obtenir

le grade de : *DOCTEUR DE L'UNIVERSITÉ DE RENNES 1*

Mention : TRAITEMENT DU SIGNAL ET TÉLÉCOMMUNICATIONS

PAR

Christophe ROSENBERGER

Équipe d'accueil : Laboratoire d'Analyse des Systèmes de Traitement de l'Information - ENSSAT

École Doctorale : Informatique, Traitement du Signal et Télécommunications

Composante Universitaire : Structure et Propriétés de la Matière

Mise en Oeuvre d'un Système Adaptatif de Segmentation d'images.

Soutenue le 7 Décembre 1999 devant la commission d'Examen

COMPOSITION DU JURY :

M.	J.-M. CHASSERY	Directeur de Recherche CNRS - Grenoble	Président du jury
Mme.	C. FERNANDEZ-MALOIGNE	Professeur à l'Université de Poitiers - ESIP	Rapporteurs
M.	A. HILLION	Professeur à l'ENSTB Brest	
M.	K. CHEHDI	Professeur à l'Université de Rennes 1 - ENSSAT	Examinateurs
M.	I.-C. LERMAN	Professeur à l'Université de Rennes 1 - IRISA	
Mme.	S. PHILIPP-FOLIGUET	Professeur à l'Université de Cergy Pontoise - ENSEA	

REMERCIEMENTS

Je tiens tout d'abord à remercier la région BRETAGNE pour son soutien financier qui m'a permis de mener à bien cette étude.

Cette thèse a été effectuée au sein du groupe Image du Laboratoire d'Analyse des Systèmes de Traitement de l'Information (LASTI) de Lannion. Aussi, je tiens à remercier Kacem Chehdi, Professeur de l'Université de Rennes I (ENSSAT) et directeur du laboratoire de m'avoir accueilli et d'avoir accepter de diriger ma thèse. Je tiens à lui exprimer ma vive gratitude pour sa disponibilité, sa rigueur scientifique et son soutien pendant ces années de recherche.

J'adresse mes sincères remerciements à monsieur J.M. Chassery, Directeur de recherche CNRS à Grenoble qui a accepté de présider mon jury de thèse.

Que madame C. Fernandez, Professeur à l'Université de Poitiers et monsieur A. Hillion, Professeur à l'École Nationale Supérieure des Télécommunications de Bretagne à Brest, trouvent ici le témoignage de ma gratitude pour avoir accepté d'examiner ce travail et d'être les rapporteurs.

Je tiens à remercier monsieur I.C. Lerman, Professeur de l'Université de Rennes I et madame S. Philipp, Professeur à l'École Nationale Supérieure de l'Électronique et de ses Applications (ENSEA) de Cergy Pontoise, pour l'honneur qu'ils m'ont fait en acceptant d'examiner ce travail.

Je tiens à rendre hommage à Claude Cariou et à Jean marc Ogier mes deux encadrants de thèse pour leurs conseils avisés et pour leur disponibilité.

Je tiens à remercier chaleureusement tous les membres du LASTI pour leur aide et pour leur convivialité. Je voudrais enfin marquer ma sympathie à l'ensemble du personnel, des étudiants et des stagiaires de l'ENSSAT qui m'ont facilité ma recherche et ont rendu mon séjour à Lannion agréable.

"La pensée n'est qu'un éclair dans la nuit mais c'est cet éclair qui est tout"

H. Poincarré.

Notations

I : une image
S : ensemble des pixels d'une image
$s = (i,j)$: un point de coordonnées (i,j) d'une image
I_s : variable aléatoire de luminance au point s de l'image I
g_i : niveau de gris de l'image
NT : nombre de pixels total d'une image
W : fenêtre d'analyse de l'image
NW : taille de la fenêtre d'analyse
$NLIG$: nombre de lignes d'une image
$NCOL$: nombre de colonnes d'une image
$G = \{0,..,NG\}$: ensemble des niveaux de gris de l'image
NG : nombre de niveaux de gris d'une image
$NCLASS$: nombre de classes d'une image
$R = \{R_1,..,R_{NCLASS}\}$ ensemble des régions de l'image
V : système de voisinage
V_s : voisinage du pixel s de l'image
$\mathbf{a} = (\mathbf{a}_1,..,\mathbf{a}_m)$: vecteur d'attributs de dimension m
$d(.,.)$: distance euclidienne
B_j : barycentre de la région R_j
N_j : nombre d'éléments de la classe R_j
Cov : matrice de covariance
μ ou E[]: moyenne
$card(A)$: nombre d'éléments d'un ensemble A
$C = \{C_1,..,C_{NCLASS}\}$: ensemble des classes d'une image
$E = \{1,..,NCLASS\}$: ensemble des étiquettes de l'image
e_s : étiquette au pixel s
P : mesure de probabilité
δ : seuil
$B(o,r)$: boule centrée au point o de rayon r
$\mathcal{O} = \{O_1,..,O_m\}$: ensemble d'individus à classer
$\mathcal{A} = \{\mathbf{a}^1,..,\mathbf{a}^m\}$: ensemble de vecteurs d'attributs
$\mathcal{P}$: partition d'un ensemble

Table des matières

Résumé **9**

Introduction **11**

Partie I : État de l'art **15**

1 État de l'art **17**
1.1 Le système visuel humain 17
1.2 Méthodes de segmentation non coopératives 18
1.2.1 Méthodes adaptées aux images uniformes 18
1.2.1.1 Détection de contours 19
1.2.1.2 Croissance de régions 20
1.2.2 Méthodes adaptées aux images texturées 21
1.2.2.1 Méthodes bayésiennes 21
1.2.2.2 Classification à partir d'attributs statistiques 29
1.2.3 Discussion 35
1.3 Méthodes de segmentation coopératives 36
1.3.1 Méthodes séquentielles 36
1.3.2 Méthodes parallèles 38
1.3.2.1 Coopération contour-région 38
1.3.2.2 Fusion de résultats de segmentation 38
1.3.3 Méthodes hybrides 40
1.3.3.1 Coopération active 40
1.3.3.2 Coopération multi-agents 42
1.3.3.3 Coopération adaptative 43
1.3.4 Discussion 44
1.4 Conclusions 44

Partie II : Système adaptatif de segmentation d'images

• **Introduction**

• **Principe du système de segmentation proposé** **47**

2 Analyse d'images **53**
2.1 Introduction 53
2.2 Analyse globale de l'image 54
2.2.1 Caractère globale de l'uniformité d'une image 54
2.2.2 Localisation des zones texturées et uniformes 58
2.2.2.1 Critère de partition des zones de l'image 60
2.2.2.2 Résultats expérimentaux 61
2.3 Analyse locale de l'image 62
2.3.1 Analyse d'attributs de texture classiques 69
2.3.1.1 Analyse de la corrélation des attributs 69
2.3.1.2 Réduction de l'espace des attributs 76
2.3.1.3 Classement de textures 78
2.3.1.4 Classification de textures 81
2.3.1.5 Conclusions 82
2.3.2 Définition de nouveaux attributs de texture 82
2.3.2.1 Définition de la texture 85
2.3.2.2 Modélisation de la fonction d'autocorrélation 1D . . 90
2.3.2.3 Calcul de la fonction d'autocovariance normalisée 1D d'une texture 91
2.3.2.4 Détermination des paramètres du modèle 92
2.3.2.5 Caractère déterministe d'une texture 93
2.3.2.6 Finesse d'une texture 94
2.3.2.7 Résultats expérimentaux 95
2.4 Conclusions 105

3 Segmentation adaptative **111**
3.1 Introduction 111
3.2 Méthode de classification automatique 111
3.2.1 Choix de la classe à subdiviser 112
3.2.2 Choix des points germes 113
3.2.3 Nuées dynamiques (partitionnement) 114
3.2.4 Évaluation d'une partition 115
3.2.5 Résultats expérimentaux 116
3.3 Segmentation 118
3.3.1 Segmentation des zones uniformes 118
3.3.2 Segmentation des zones texturées 120
3.4 Résultats expérimentaux 121
3.5 Conclusions 121

4 Fusion de résultats de segmentation **127**
4.1 Introduction 127
4.2 Les algorithmes génétiques 127
4.3 Critère d'évaluation d'un résultat de segmentation 129
4.3.1 Définition du critère 129
4.3.2 Disparité intraclasse 131
4.3.2.1 Régions uniformes 131
4.3.2.2 Régions texturées 131
4.3.3 Disparité interclasse 133
4.3.3.1 Disparité entre régions de type différent 134
4.3.3.2 Disparité entre régions de même type 134
4.3.4 Validation du critère 135
4.3.5 Conclusions 138
4.4 Méthode de fusion 139
4.5 Résultats expérimentaux 140
4.6 Conclusions 141

• Propriétés du système développé **145**

Conclusion **149**

Annexes **153**

A Descripteurs de texture **155**
A.1 Attributs de la matrice de cooccurrence 155
A.2 Attributs de la matrice de longueur de plages 158
A.3 Attributs des histogrammes locaux du premier ordre 159
A.4 Attributs des extrema locaux 159
A.5 Attributs de l'intégrale curviligne 160
A.6 Moments statistiques 160

Bibliographie **161**

Liste des tableaux **166**

Liste des figures **167**

Résumé

Le traitement d'images suscite un intérêt croissant à mesure que l'image s'impose comme un support et une source d'informations privilégiés. La qualité de l'interprétation d'une image dépend fortement de celle de la segmentation. Malgré la grande diversité de méthodes, les résultats de segmentation restent moyens et varient beaucoup en fonction de la technique choisie. Une méthode de segmentation générale et automatique est difficile à concevoir étant donnés les différents types de régions pouvant être présentes dans une image. Afin de contribuer à résoudre ce problème, nous proposons un système adaptatif de segmentation d'images.

Après un travail de synthèse permettant de répertorier les différentes méthodes de segmentation existantes en fonction du type d'informations pour lesquelles elles sont performantes, un système original de segmentation est proposé. L'originalité de ce système réside dans l'adaptation des traitements au contexte local de l'image avec le minimum de connaissances *a priori*. Il est constitué de trois modules de traitement. *Le premier module* permet d'analyser finement l'image à deux niveaux. Le premier niveau identifie d'une part, le contexte global de l'image à traiter (image majoritairement composée de régions uniformes ou texturées) afin d'adapter la suite des traitements et, distingue d'autre part, les zones texturées et uniformes la composant. Le second niveau du module concerne l'analyse locale de l'image à segmenter afin de caractériser chacune des régions détectées par des attributs classiques de textures pertinents (obtenus par analyse statistique) et des attributs que nous avons définis. Ces paramètres complémentaires ont été déterminés à partir d'un modèle de texture basé sur la décomposition de Wold de la fonction d'autocovariance. Ils permettent d'obtenir des informations sur le type de texture (aléatoire ou déterministe) et sur sa granularité (grossière ou fine). Cette analyse plus fine d'une région texturée permet, d'une part, de faciliter le choix de la méthode de segmentation appropriée et, d'autre part, d'adapter la taille du support d'analyse de la région à segmenter. *Le deuxième module* déclenche la méthode de segmentation adaptée au contexte local de l'image en utilisant une méthode de classification non supervisée que nous avons développée. Enfin, *le troisième module* permet de fusionner soit les résultats de plusieurs méthodes de segmentation d'une même image, soit les résultats de segmentation de chaque bande dans le cas d'une image multi-composantes. La méthode de fusion développée adopte une approche génétique en combinant les résultats de segmentation pondérés par un critère d'évaluation. Le système a été validé sur différents types d'image (synthétiques et réelles de télédétection).

Mots clef : segmentation adaptative, analyse d'images, modèle de textures, algorithme génétique, fusion, critère d'évaluation d'un résultat de segmentation.

Introduction

Parmi les perspectives offertes par l'informatique, le traitement d'images suscite un intérêt croissant à mesure que l'image s'impose comme un support et une source d'informations privilégiés. La multiplication des dispositifs d'acquisition conduit à la production d'un grand nombre d'images, qu'il devient difficile d'exploiter manuellement. Le traitement d'images s'inscrit alors dans un processus préliminaire destiné à préparer les images à leur interprétation, leur stockage et leur transmission. Une image, représentation 2D d'une scène 3D et provenant d'un capteur, est dans un premier temps numérisée afin de pouvoir être traitée par un ordinateur. A partir d'une image numérique, il convient de la traiter, d'y extraire des informations puis de l'interpréter. Trois étapes sont donc nécessaires :

- **amélioration de la qualité de l'image :** il s'agit ici de corriger d'éventuels défauts introduits par le capteur lors de l'acquisition de l'image, de l'environnement ou des conditions d'acquisition. Plusieurs traitements sont possibles :
  - rehaussement : amélioration du contraste d'une image dans le cas de scènes sous-exposées,
  - filtrage : suppression du bruit contenu dans l'image. Ce bruit provient de perturbations pendant la numérisation de l'image ou pendant son acquisition,
  - restauration : suppression du flou et éventuellement du bruit dans l'image. Le flou peut provenir d'un mouvement du capteur lors de l'acquisition.
- **segmentation :** ce traitement consiste à partitionner une image I en sous-ensembles R_i, appelés régions tels que :

$$\begin{gathered} \forall i,\ R_i \neq \emptyset \\ \forall i \neq j,\ R_i \cap R_j = \emptyset \\ I = \cup_i R_i \end{gathered}$$

  Une région est un ensemble connexe de points de l'image ayant des propriétés communes qui les différencient des pixels des régions voisines.
- **interprétation :** cette phase consiste à exploiter le résultat des traitements précédents afin d'obtenir des informations de haut niveau sur l'image.

Ainsi, si l'on considère l'exemple d'une image de type télédétection représentant des surfaces cultivées, l'amélioration de la qualité de l'image acquise peut porter sur la correction du flou engendré par la traversée de la couche atmosphérique (restauration). La segmentation aura pour but de différencier les champs de cultures différentes. L'interprétation guidée par quelques connaissances *a priori* permettera par exemple d'identifier les cultures de blé, d'orge ou d'extraire d'autres

informations.

Position du problème :

La segmentation d'images est une étape primordiale en traitement d'images puisqu'elle conditionne la qualité de l'interprétation. Si l'on se réfère à la littérature dans ce domaine, on se rend compte que ce problème est difficile et qu'il est loin d'être résolu. Beaucoup de méthodes existantes donnent de bons résultats mais elles se restreignent à un type d'image particulier ou nécessitent des connaissances *a priori* qui sont dans la majorité des cas difficilement accessibles.

Pour pallier ces inconvénients, plusieurs approches coopératives apparaissent afin de combiner plusieurs méthodes de segmentation. Nous en distinguons trois types :

- *l'approche séquentielle* : elle consiste à combiner plusieurs méthodes de telle façon que le résultat de la segmentation d'une image par une technique serve de point de départ à une autre. Un exemple désormais classique de ce type de méthode est l'approche contour-région [FBC94],[BW91],
- *l'approche parallèle* : la technique de coopération parallèle consiste à exploiter les résultats issus de plusieurs segmentations en fusionnant ceux-ci ou en utilisant leurs complémentarités [FJ91], [fB95],
- *l'approche hybride* : il s'agit de combiner les deux approches précédentes [SGC95],[SP95],[Ker97].

L'inconvénient majeur de l'approche séquentielle est sa faible tolérance à l'échec d'une méthode de segmentation dans la chaîne de traitements. Par ailleurs, elle est globale et donc ne peut pas être adaptative. La seconde technique de coopération permet effectivement de s'affranchir de ce désavantage. Néanmoins, l'intégration des différents résultats est délicate dans la mesure où, à la fois la complémentarité et la redondance des méthodes employées doivent être prises en compte. De plus, la fusion de résultats par exemple, n'a de sens que lorsque les méthodes utilisées sont adaptées au type de l'image. La troisième approche tente de pallier les inconvénients des deux précédentes, ce qui rend sa conception très difficile. D'autre part, la remise en cause d'un résultat exploitée dans cette approche n'a d'intérêt que si l'on est capable de tirer des conclusions sur l'échec d'une méthode étant donnée une image.

Le problème majeur de ces approches est leur incapacité à s'adapter au contexte.

Objectif :

L'objectif fixé dans le cadre de cette thèse est le développement d'un système de segmentation permettant d'adapter localement les traitements d'une image étant donné son contexte. On peut en effet espérer une meilleure efficacité en adaptant un traitement aux données plutôt qu'en appliquant un seul outil général. Dans le cas

où plusieurs méthodes sont adaptées à un type de régions, les différents résultats de segmentation, potentiellement corrects, sont fusionnés dans une seconde étape.

Ce mémoire de thèse se décompose donc en deux parties :

- la première partie concerne un travail de synthèse portant sur la segmentation d'images,
- la seconde partie présente le système de segmentation que nous proposons. Nous détaillons les trois modules constituant ce système, sa réalisation et sa validation au travers de plusieurs résultats expérimentaux :
  - le premier module concerne l'analyse de l'image à traiter à deux niveaux, global et local. Le premier niveau permet d'identifier le type de l'image à analyser (texturée ou uniforme) et de répertorier les zones de l'image en deux classes (région texturée et uniforme ou faiblement texturée). Le second niveau caractérise ces régions par des attributs en vue d'adapter la méthode de segmentation à employer,
  - le deuxième module exploite les attributs extraits dans le premier module pour segmenter l'image en utilisant une méthode de classification non supervisée que nous avons développée,
  - le troisième module permet de traiter des images multi-composantes en fusionnant les résultats de segmentation de chacune d'entre elles.

Nous concluons ce mémoire et nous proposons quelques perspectives.

PARTIE I

État de l'art en segmentation d'images

Chapitre 1

État de l'art

L'objectif de l'état de l'art est de pouvoir analyser les principales méthodes de segmentation existantes afin de mettre en avant leurs avantages et inconvénients, et de sélectionner celles qui vont être intégrées dans le système de segmentation que nous cherchons à concevoir. Beaucoup de travaux ont été consacrés au problème de la segmentation. Chronologiquement, ce sont les méthodes adaptées aux images uniformes au sens du niveau de gris et faiblement bruitées qui sont apparues les premières (détection de contours, croissance de régions). Plusieurs méthodes d'analyse de textures ont ensuite permis de segmenter des images texturées [HSD73]. Etant donnée la complexité du problème, les approches récentes tentent de faire coopérer plusieurs méthodes de segmentation, afin soit de combiner les avantages de chaque méthode, soit d'exploiter leurs complémentarités. Ces méthodes plus orientées vers une approche système essaient de s'affranchir d'heuristiques pour segmenter une image, dans le but d'étendre leur domaine d'application à un plus large éventail de type d'images.

Nous présentons dans un premier temps les propriétés du système visuel humain qui reste le modèle de référence dans la conception de beaucoup de méthodes de segmentation d'images. Nous présentons ensuite un état de l'art en segmentation d'images, en distinguant les méthodes non coopératives et coopératives et nous concluons.

1.1 Le système visuel humain

La vision active est une méthodologie de conception d'un système de vision par ordinateur. La connaissance du système visuel humain peut se révéler essentielle lors de l'élaboration d'un tel système.

L'être humain est un observateur actif car il est capable de modifier ses paramètres visuels afin d'acquérir des informations sur l'environnement, utiles pour la tâche fixée. Ainsi, lorsque l'homme regarde une scène, ses yeux ajustent le

niveau d'éclairement, convergent ou divergent et sa tête se positionne afin d'avoir un meilleur point de vue. Un système de vision active essaie de reproduire cette faculté d'adaptation.

L'interprétation d'une image est en général une opération relativement simple pour un observateur. Elle fait intervenir différents processus qui vont de l'acquisition d'images, selon les différents modes de perception (vision centrale ou périphérique), à l'activation de processus cognitifs qui permettent à l'observateur, en fonction de sa culture, de structurer son observation par rapport à ses connaissances. La compréhension d'une image n'est pas innée chez l'homme, s'il ne possède pas de connaissances sur les éléments constituant l'image, il risque de ne pas pouvoir les identifier.

La vision humaine peut être considérée comme un système général au niveau de ses mécanismes de compréhension et de raisonnement appropriés. Ces mécanismes permettent l'interprétation instantanée de scènes très complexes et variées. Ils font appel à des connaissances *a priori* de ce que nous observons, comme les modèles et les catégorisations des objets, et à la maîtrise des stratégies de commande de nos capteurs visuels, comme la fixation, la focalisation et le suivi d'objets.

Beaucoup de systèmes de traitement d'images tentent de reproduire la capacité du système visuel humain à adapter un traitement au contexte, à faire coopérer plusieurs tâches tout en exploitant des connaissances. Cette volonté de "copier" le système visuel humain se retrouve dans la plupart des systèmes de segmentation existants comme nous allons le montrer.

1.2 Méthodes de segmentation non coopératives

Nous présentons ici les différents types de méthodes de segmentation non coopératives ne nécessitant pas ou peu de connaissances *a priori*. Ces méthodes sont les plus classiques en segmentation d'images.

Nous pouvons distinguer deux types de méthodes de segmentation non coopératives en fonction du type d'images pour lesquelles elles sont adaptées :

- méthodes adaptées aux images uniformes,
- méthodes adaptées aux images texturées.

1.2.1 Méthodes adaptées aux images uniformes

La segmentation des images uniformes consiste soit à détecter une variation entre deux régions d'une image, soit à regrouper les pixels ayant des caractéristiques proches. L'information sur le niveau de gris d'un pixel est une information suffisante

pour réaliser cette tâche. La première approche concerne la détection de contours et la seconde fait référence à la croissance de régions.

1.2.1.1 Détection de contours

Nous exposons dans ce qui suit les différentes méthodes de détection de contours.

a) Les méthodes dérivatives [CP95]

Le principe des méthodes dérivatives repose sur la détection d'une transition d'un signal à partir de sa dérivée d'ordre 1 ou 2.

Parmi les opérateurs du premier ordre, on trouve le gradient, le masque de Prewit, de Sobel, de Kirsh [Kir71], [Sob78]... Ces opérateurs sont sensibles au bruit et nécessitent l'utilisation d'un seuil heuristique. D'autres opérateurs MDIF et NAGDIF ont été développés [CP95]. Ils combinent des opérateurs de lissage et de dérivation. Ils donnent de bons résultats en milieu peu bruité avec des transitions franches et ont l'avantage d'être rapides en temps de calcul.

Les opérateurs du deuxième ordre localisent les contours en déterminant les passages par zéro du laplacien.

Marr et Hildreth suggèrent [MH80], par similitude avec l'oeil humain, l'utilisation d'un opérateur Laplacien-Gaussien. La méthode proposée consiste à convoluer l'image par une gaussienne avant d'appliquer un filtre Laplacien. Ils proposent l'utilisation de plusieurs masques correspondant à différentes valeurs de la variance σ^2 de la gaussienne. Les masques de variance trop large dégradent l'image et ne permettent pas la détection des structures fines, à l'inverse, une variance trop faible conduit à une détection du bruit trop importante. La localisation par ce type de méthode n'est pas très précise et dans le cas d'images complexes, les résultats obtenus sont très sensibles au bruit et engendrent souvent de fausses détections. En effet, des faux contours sont détectés dans les zones bruitées ou texturées (même légèrement) et des contours significatifs dont le gradient est faible sont oubliés.

Pour surmonter ces difficultés de précision de localisation et d'efficacité de détection, Canny [Can86] propose de formaliser mathématiquement une approche optimale de détection. Pour déterminer le filtre optimal, il définit trois critères : bonne détection, bonne localisation et non multiplicité des réponses. Trois filtres optimaux majeurs ont été obtenus en prenant en compte ces trois critères : le filtre de Canny [Can86], le filtre de Deriche [Der87] et le filtre de Shen et Castan [JS92], [CZS89]. L'avantage du filtre de Deriche est qu'il est possible de l'implémenter de façon récursive.

Il existe dans la littérature d'autres opérateurs de détection de contours. On peut citer par exemple les opérateurs hyperboliques. Un exemple de ce type d'approche est donné dans [Yen86]. L'opérateur hyperbolique utilisé donne de meilleurs résultats que ceux énoncés précédemment et peut être implémenté de façon récursive.

b) Méthodes de détection par une approche variationnelle

Une autre approche d'extraction de contours proposée utilise le modèle de contours actifs "snakes" [KWT88]. Les contours sont modélisés par des courbes élastiques qu'il s'agit d'ajuster au mieux. Pour cela, on associe une énergie dont la définition est étroitement liée à celle des points contours. Les minima locaux de cette énergie correspondent aux contours s'ajustant le mieux au modèle utilisé [PD99].

1.2.1.2 Croissance de régions

La croissance de régions consiste à fusionner un ensemble de pixels ayant des caractéristiques proches. Elle peut se faire à partir de plusieurs niveaux, le plus élémentaire étant le pixel.

a) Agrégation de pixels

La méthode d'agrégation de pixels est une méthode ascendante qui permet de regrouper un ensemble de pixels selon un double critère d'homogénéité et d'adjacence. Le processus peut être itératif en regroupant les pixels ayant un niveau de gris proche de points germes [CL94], [CJA96]. Ces méthodes donnent de bons résultats même si la localisation des contours est moins précise que par les techniques de détection de contours. De plus, les seuils d'agrégation influent beaucoup sur la qualité du résultat.

Une variante d'agrégation de pixels consiste à regrouper les pixels ayant des niveaux de gris proches soit par une méthode de multi-seuillage soit par classification d'attributs tels que la moyenne ou la variance. Ces méthodes peuvent intégrer des informations locales (histogramme local [KCC95] ou attributs sur un voisinage de petite taille) les rendant assez robustes à la présence de bruit.

b) Relaxation

La relaxation est une technique permettant de faire évoluer une partition de l'image en régions en n'intégrant que deux types d'informations : un attribut du pixel traité et la classe de ses voisins. Le processus est itératif et converge vers une partition de l'image à partir d'une pré-segmentation de celle-ci ou d'un tirage aléatoire. Cette technique est utilisée en segmentation d'images uniformes en considérant des histogrammes locaux ou des moments sur une fenêtre de petite taille (par exemple 3×3 pixels) [HS89], [HH97]. Une technique similaire a été

utilisée dans [Buf89] en utilisant cette fois une mesure de similarité spectrale obtenue à partir de filtres de Gabor au lieu des histogrammes locaux.

Cette approche peut produire un phénomène de sur-segmentation lorsque les régions sont de taille importante. Cette technique peut être aussi utilisée afin d'améliorer un résultat de segmentation.

1.2.2 Méthodes adaptées aux images texturées

On peut décomposer l'ensemble des méthodes adaptées aux images texturées en deux classes disjointes. La première concerne les méthodes bayésiennes, c'est à dire les méthodes utilisant une modélisation aléatoire de la texture. Le second type consiste à extraire des attributs statistiques d'ordre 1, 2 ou 3 puis à utiliser une méthode de classification (méthodes d'analyse de données, méthodes neuronales,..).

1.2.2.1 Méthodes bayésiennes

Nous présentons ici les différentes méthodes de segmentation bayésiennes. Soit $I = \{I_s,\ s \in S\}$ un champ de variables aléatoires (les observations) indexées par une grille S de NT sites s, chacune de ces variables aléatoires étant à valeurs dans $\mathbb{R}$ codées sur $G = \{0,..,NG\}$. Une réalisation g_s de la variable aléatoire I_s est le niveau de gris de l'image au site s.

De même, soit $C = \{C_s,\ s \in S\}$ un champ de variables aléatoires (les classes) sur la grille S, à valeurs dans $E = \{1,..,NCLASS\}$. Une réalisation e_s de la variable aléatoire C_s est l'étiquette du pixel situé au site s.

a) Echantillonnage

Nous rappelons brièvement la définition d'un champ de Markov : Le champ aléatoire $I = \{I_s,\ s \in S\}$, relatif à un système de voisinage V, est un champ aléatoire de Markov discret si sa distribution de probabilité P satisfait les trois conditions suivantes :

1. Positivité : $P[I = g] > 0,\ \ \forall g \in G$
2. Markovianité : $P[I_s = g_s / I_t = g_t,\ t \in S - \{s\}] = P[I_s = g_s / I_{Vs} = g_{Vs}]$ où Vs est le voisinage associé au point s.
3. Homogénéité : La probabilité conditionnelle $P[I_s = g_s / I_{Vs} = g_{Vs}]$ dépend seulement de la configuration des voisins du site s et est invariante par translation conservant la même configuration du voisinage.

Un champ de Gibbs discret a une distribution de probabilité sous la forme [Bes74]:

$$\pi_t = (g) = \frac{1}{Z} e^{\frac{-U(g)}{t}} \tag{1.1}$$

Le paramètre t représente la température absolue, la fonction $U(g)$ est une fonction d'énergie (avec $U(0) = 0$) et Z est une constante de normalisation. L'équivalence entre les champs de Markov et les champs de Gibbs fournit une formulation explicite de la distribution $P[I = g]$.

Nous donnons maintenant plusieurs algorithmes d'échantillonnage suivant une loi de Markov utilisés dans les méthodes bayésiennes de segmentation.

L'algorithme de Métropolis [MRR+53] permet d'obtenir un échantillon $(g_1,..,g_n)$ d'une distribution de Gibbs. Une variante de cet algorithme est proposée dans [KZB92] où la température est abaissée progressivement. Cette approche permet d'obtenir de meilleurs résultats en terme d'erreur de classification par rapport au métropolis classique avec une complexité raisonnable.

L'échantillonneur de Gibbs présente l'avantage de générer une chaîne de Markov ayant la propriété d'ergodicité [GG84]. L'hypothèse d'ergodicité consiste à considérer que les caractéristiques d'ensemble sont équivalentes aux caractéristiques statistiques spatiales.

Le recuit simulé permet de construire une chaîne de Markov ayant une matrice de transition dépendant de la température T en faisant tendre progressivement celle-ci vers 0 [GG84]. De par sa nature stochastique, cette procédure échappe aux maxima locaux.

b) Classification bayésienne supervisée

On définit les probabilités suivantes :
- $P[C = e]$: probabilité a priori d'étiquetage $[C = e]$ où $e = \{e_s \in E, s \in S\}$.
- $P[I = g/C = e]$: probabilité conditionnelle d'avoir $[I = g]$ sachant $[C = e]$ où $g = \{g_s \in G, s \in S\}$.
- $P[C = e/I = g]$: probabilité a posteriori d'avoir $[C = e]$ sachant la réalisation $[I = g]$.
- $P[C_s = e_s/I = g]$: probabilité marginale a posteriori d'avoir $[C_s = e_s]$ sachant l'événement $[I = g]$.

Il est possible de calculer cette dernière à partir des autres en utilisant la règle de Bayes :

$$P[C_s = e_s/I = g] = \frac{P[I = g/C_s = e_s].P[C_s = e_s]}{\sum_{t \in S} P[I = g/C_t = e_t]P[C_t = e_t]} \tag{1.2}$$

Si l'on se place dans le cadre où les deux premières lois citées précédemment sont connues, on est alors en mesure de calculer pour chaque pixel sa probabilité d'appartenir à une classe étant donnée une observation. Les méthodes employées sont donc supervisées par la connaissance a priori des distributions. Les méthodes

d'attribution d'une classe à un pixel en mode supervisé sont les suivantes.

b.1) Règle de décision bayésienne générale

Le théorème de Bayes permet de calculer la probabilité a posteriori $P[C_s = e_s/I_k = g_k]$ de la façon suivante :

$$P[I_k = g_k/C_s = e_s] = \frac{P[I_k = g_k/C_s = e_s]P[C_s = e_s]}{\sum_{t \in S} P[I_k = g_k/C_t = e_t]P[C_t = e_t]}$$

On décide que le pixel de l'image situé au site k a l'étiquette e_i si et seulement si : $P[C_k = e_i/I_k = g_k] \ > \ P[C_k = e_j/I_k = g_k] \ \forall \ j \neq i$.

b.2) Maximum a posteriori (MAP)

Le principe de la méthode MAP consiste à déterminer les étiquettes $e^* = (e_1,..,e_N)$ des pixels de l'image à partir de l'ensemble des niveaux de gris telles que :

$$e^* = arg \max_{e \in E^S} \ P[C = e/I = g]$$

Il n'est pas possible de parcourir l'ensemble de E pour déterminer la configuration maximisant cette probabilité. Une solution consiste à utiliser l'algorithme du recuit simulé via l'échantillonneur de Gibbs pour approcher MAP si la distribution *a priori* de C est de Gibbs.

Quand $T \rightarrow 0$, on peut remarquer que la probabilité π_T est concentrée sur l'estimateur du MAP de e^* ou bien que π_T est uniformément distribuée sur l'ensemble des maxima globaux.

Le recuit simulé permet de construire une chaîne de Markov ayant une matrice de transition dépendant de la température T en faisant tendre progressivement celle-ci vers 0. Le schéma de "refroidissement" par analogie en thermodynamique, permet d'assurer la convergence du procédé vers un maximum global. Cette solution donne une approximation de l'estimateur MAP.

Une méthode d'estimation de MAP peut être trouvée dans [RC91]. Cette approximation est réalisée par la minimisation d'une fonction d'énergie donnant lieu à deux nouvelles méthodes GRAD (gradient) et GMAG (gradient-magnitude). Une approche assez similaire est proposée dans [WZX92]. Une méthode basée sur une approche multi-échelles [BS92] permet une estimation itérative de l'estimateur MAP à plusieurs résolutions de l'image. Elle est appelée SMAP (sequential MAP) et donne de meilleurs résultats que l'estimateur MAP mais nécessite plus de calculs.

Une autre méthode hiérarchique peut être trouvée dans [MCG92].

b.3) Maximisation des marginales a posteriori (MPM)

Le principe de la méthode MPM consiste à déterminer la classe $e_s^* \in E$ du pixel s de l'image à partir de l'ensemble des niveaux de gris telle que [Bes74] :

$$e_s^* = arg \max_{e_s \in E} \; P[C_s = e_s / I = g]$$

L'obstacle majeur de cette méthode est le calcul de ces probabilités marginales a posteriori. Néanmoins, si l'on suppose que cette distribution est de Gibbs, alors il est possible d'approximer ces lois marginales. La méthode consiste à estimer la distribution a posteriori par une méthode de Monte-Carlo, générant une chaîne de Markov ayant comme distribution limite la probabilité π_T définie en (1.1). La distribution marginale a posteriori s'écrit donc sous la forme :

$$P[C_s = e_s / I = g] \simeq \frac{1}{n - n_0} \sum_{t=n_o+1}^{n} 1_{\{e_s^{(t)} - e_s\}}$$

où n_0 est le nombre d'échantillons nécessaires pour assurer la stabilité et n la longueur de la chaîne considérée. Cette approximation correspond à la fréquence d'apparition de l'étiquette e_s sur le site s, calculée à partir de l'échantillon $(e_s^{(n_0)},..,e_s^{(n)})$.

b.4) Maximisation des conditionnelles a posteriori(ICM)

Le principe de la méthode ICM (iterated conditional mode) est le suivant : On définit $\hat{e}$ l'estimée de e^* l'ensemble des étiquettes de l'image originale. L'estimation de e^* est réalisée de façon itérative :

$$\hat{e_s}^{(n+1)} = arg \max_{e_s \in E} P[C_s = e_s / \hat{C_t}^{(n)} = \hat{e_t}^{(n)}, t \in V_s - s, I_s = g_s]$$

où V_s est un voisinage du pixel situé en s.

La configuration initiale pour chaque pixel s est la classe $\hat{e_s}^{(0)}$ telle que :

$$\hat{e_s}^{(0)} = arg \max_{e_s \in E} P[I_s = g_s / C_s = e_s]$$

La convergence de l'algorithme est atteinte en pratique au bout de 5 à 10 itérations. Un version modifiée du ICM a été proposée dans [CTBP95] en utilisant une implémentation multi-échelles permettant une meilleure robustesse vis-à-vis du bruit.

b.5) Évaluation des performances des ces méthodes

En ce qui concerne la complexité de mise en oeuvre, les méthodes MAP et MPM se révèlent être moins intéressantes. En effet, la capacité de mémoire requise dans le cas de MPM est importante car pour m classes, il est nécessaire de stocker $(m-1).card(S)$ nombres. De plus, l'utilisation du recuit simulé est un désavantage de la méthode MAP. A contrario, la méthode ICM se distingue des deux précédentes par sa simplicité de mise en oeuvre et pour son coût-mémoire économique.

Les trois méthodes sont performantes lorsque l'image n'est perturbée que par un bruit blanc. En pratique, la méthode ICM donne des résultats un peu moins bons. La taille du voisinage dans l'estimation du label d'un pixel dans la méthode ICM est primordiale. En effet, l'amélioration apportée en prenant en compte 8 voisins plutôt que 4 est significative.

On peut aussi citer d'autres méthodes d'optimisation comme le champ moyen TPM (Threshold Posterior Mean). La méthode LH (Lagrange-Hopfield) détaillée dans [LS96] permet de maximiser la vraisemblance en utilisant des techniques d'optimisation. On obtient de meilleurs résultats que ICM mais au prix d'un nombre d'itérations beaucoup plus élevé (environ 180 fois supérieur).

c) Classification bayésienne non supervisée

Dans le cadre non supervisé, les distributions a priori du champ aléatoire modélisant une image ne sont pas connues. Il faut donc une étape préliminaire afin d'estimer les paramètres relatifs à ces distributions. Il y a deux types de paramètres : On note θ les paramètres relatifs à la distribution π et ϕ ceux modélisant le bruit. L'estimation des paramètres $\Theta = (\theta,\phi)$ peut être réalisée de deux façons différentes :

- soit on dispose de données d'apprentissage auquel cas, on estime les paramètres θ et ϕ puis on segmente en utilisant une méthode citée précédemment,
- quand il n'y a pas de données d'apprentissage, l'estimation de θ et ϕ fait partie intégrante de la segmentation.

Nous explicitons par la suite chacune des deux méthodes.

c.1) Estimation à partir de données d'apprentissage

Ce cas est le plus simple puisque l'on dispose d'observations relatives à une scène donnée. Le point essentiel est la détermination des paramètres θ puisqu'on obtient ϕ facilement en maximisant la densité conditionnelle d'une observation sachant sa configuration. Il existe deux méthodes principales d'estimation de θ :

- la méthode du maximum de vraisemblance : Elle consiste à maximiser la distribution π du champ des classes. On utilise un algorithme du gradient pour

déterminer θ. Cette méthode ne peut fonctionner que si les modèles utilisés sont simples comme les modèles d'Ising ou gaussiens,
- la méthode du pseudo-maximum de vraisemblance : On maximise ici le produit suivant par l'algorithme du gradient

$$\prod_{s \in S} \pi_\theta[e_s/e_t,\ t \in V_s]$$

c.2) Méthodes générales d'estimation

L'estimation bayésienne consiste à déterminer à partir d'observations $B \in S$ l'ensemble des labels sur un ensemble $A \subseteq B$ tel que la probabilité sachant les observations I_B est maximale.

Il existe deux types d'approche d'estimation de paramètres en segmentation bayésienne :
- maximisation de la vraisemblance : on approche le maximum de vraisemblance par des méthodes itératives comme les algorithmes EM ou SEM,
- estimation itérative à partir de l'espérance conditionnelle : la méthode ICE consiste à approcher les paramètres via l'espérance conditionnelle.

c.2.1) Maximum de vraisemblance

On essaie de maximiser la vraisemblance $L_\Theta(g)$ d'un échantillon $g = (g_1,..,g_N)$ définie par [You89] :

$$L_\Theta(g) = \prod_{j=1}^{N} (\sum_{i=1}^{n} f(g_j,\phi_i))$$

où $\Theta = (\theta = (\pi_1,..,\pi_n),\phi = (\phi_1,..,\phi_n))$, $n = card(E)^{cardB}$ et $f(g_j,\phi_i)$ est la densité conditionnelle d'avoir $[I_j = g_j]$ sachant que $[C_j = e_i]$.

c.2.2) Algorithme EM

L'échantillon g peut être considéré comme la partie connue de la donnée incomplète $y = (g,E)$ où E est l'ensemble des labels. Le principe de l'algorithme EM est de maximiser en $\Theta = (\theta,\phi)$ la log-vraisemblance $\log L_\Theta(g)$ des données incomplètes observées g [DLR77]. Celle-ci s'écrit sous la forme :

$$L_\Theta(y) = k_\Theta(y/g) L_\Theta(g)$$

où k_Θ est la densité conditionnelle de y sachant g. On définit par ailleurs la fonction suivante à partir de l'estimée $\Theta^{(t)}$ à l'itération $t > 1$:

$$Q(\Theta/\Theta^{(t)}) = E[log L_\Theta(y)/g,\Theta^{(t)}]$$

La propriété fondamentale de l'algorithme EM est l'exploitation du fait que la log-vraisemblance est monotone croissante vers une limite. D'autre part, si la solution initiale est éloignée de la solution limite, la convergence est lente. Plusieurs variantes de cet algorithme ont été proposées notamment dans [LMMZ92]. Langan et al ont approximé le calcul de la vraisemblance par l'intermédiaire du calcul des densités conditionnelles pour accélérer le processus et pour améliorer la qualité des résultats.

c.2.3) Algorithme SEM

Cet algorithme est une variante de la méthode EM auquel on a rajouté une étape probabiliste stochastique. Le déroulement est donc le suivant :

- Initialisation : soit un échantillon $g = (g_1,..,g_N)$, on choisit de façon aléatoire les probabilités initiales d'appartenance à une classe E ($NCLASS$ étant un majorant fixé du nombre de classes). On obtient donc $p_i^{(0)}(g_j)$, $i = 1,..,NCLASS$. On se donne en plus un seuil $\delta(N,n)$ compris entre 0 et 1.
- Etape stochastique : on tire de façon aléatoire pour chaque point g_j une réalisation de la variable multinomiale d'ordre 1 suivant $(p_i^{(t)}(g_j),\ j = 1,..,n)$. Les réalisations $e_i^{(t)}(g_j)$ forment une partition $E^{(t)}$ de l'échantillon g tel que :

$$e_i^{(t)} = \{g_j / e_i^{(t)}(g_j) = 1\}$$

  Si il existe $i \in \{1..n\}$ tel que $card(E_i^{(t)}) < N.\delta$ alors on revient à l'étape d'initialisation sinon on passe à l'étape suivante.
- Etape maximisation : estimation par le maximum de vraisemblance des paramètres Θ.
  - estimation de $\theta^{(t)}$: $\pi^{(t)} = \frac{1}{N}\sum_{j=1}^{N} e_i^{(t)}(g_j)$
  - estimation de $\phi^{(t)}$: On estime $\phi_i^{(t)}$ à partir de la moyenne $\mu_i^{(t-1)}$ et de la matrice de covariance $Cov_i^{(t-1)}$ de $\phi_i^{(t-1)}$.

$$\mu_i^{(t)} = \frac{\sum_{j=1}^{N} g_j.e_i^{(t)}(g_j)}{\sum_{j=1}^{N} e_i^{(t)}(g_j)}$$

$$Cov_i^{(t)} = \frac{\sum_{j=1}^{N}(g_j - \mu_i^{(t)})^t(g_j - \mu_i^{(t)})e_i^{(t)}(g_j)}{\sum_{j=1}^{N} e_i^{(t)}(g_j)}$$

- Etape expectation : on calcule les probabilités suivantes à partir de $\Theta^{(t)}$

$$p_i^{(t)} = \frac{\pi_i^{(t)}.f(g_j,\phi_i^{(t)})}{\sum_{i=1}^{n} \pi_i^{(t)}.f(g_j,\phi_i^{(t)})}$$

  où $f(g_j,\phi_i)$ est la densité conditionnelle d'avoir $[I_j = g_j]$ sachant que $[C_j = e_i]$.

Les avantages de la méthode SEM sont la vitesse de convergence plus rapide que l'algorithme EM et une connaissance a priori du nombre de classes non nécessaire. Cette technique a montré son efficacité en segmentation de textures non supervisée [CC96].

c.2.4) Algorithme ICE

La méthode ICE est basée sur l'approximation du calcul des espérances conditionnelles $E[\theta/I]$ et $E[\phi/I]$. Une itération de l'algorithme est donnée par :

- Initialisation : fixer $\Theta^{(0)} = (\theta^{(0)},\phi^{(0)})$
- Calcul de $\Theta^{(t)}$ à partir de $\Theta^{(t-1)}$:

$$\theta^{(t)} = E_{\theta^{(t)}}[\theta^*/O] \simeq \frac{1}{M}\sum_{k=1}^{M}\theta^*(\hat{e}^{(k)})$$

$$\phi^{(t)} = E_{\theta^{(t)}}[\phi^*/O] \simeq \frac{1}{M}\sum_{k=1}^{M}\phi^*(\hat{e}^{(k)},g)$$

où $\hat{e}^{(1)},..,\hat{e}^{(M)}$ sont des réalisations indépendantes de E obtenues par échantillonnage de la distribution a posteriori de E sachant $[I = g]$ calculée sur la base de $\Theta^{(t-1)}$.

Le principe de la procédure ICE est conceptuellement différent de l'algorithme EM puisque la notion de vraisemblance n'intervient pas. Cependant, cette approximation stochastique se révèle être assez coûteuse. Il existe néanmoins quelques alternatives utilisant une approximation déterministe de l'espérance conditionnelle. On trouve dans [ABHP92] une alternative intéressante mais plus coûteuse. En effet, chaque itération de l'algorithme suggéré couple la méthode ICE avec le recuit simulé. Une comparaison entre cette approche et l'algorithme de Younes montre son efficacité.

c.2.5) La relaxation probabiliste

Le principe général de la méthode consiste à définir une probabilité a priori pour chaque pixel d'appartenir à une classe donnée. Ces probabilités sont renforcées ou affaiblies suivant la compatibilité des décisions entre elles. Après quelques itérations, on obtient la convergence des probabilités et il ne reste plus qu'à effectuer un seuillage de celles-ci pour la segmentation [HS89].

On pose $I = \{I_1,..,I_N\}$ une image donnée et $e = \{e_1,..,e_N\}$ l'ensemble des étiquettes de l'image. On formalise le problème de la manière suivante :

- $p_s(e_j)$: probabilité a priori que le pixel s appartienne à la classe e_j.
- $P_s = (p_s(e_1),.,p_s(e_j),.,p_s(e_N))^T$: vecteur de probabilité a priori du pixel s.

– On définit le coefficient $r_{ij}(k,m)$ mesurant la compatibilité entre l'événement $[C_i = e_k]$ et $[C_j = e_m]$.

On suppose que la densité conditionnelle à l'observation de chaque classe est une gaussienne de dimension d. Le processus de relaxation est le suivant :

1. estimation des paramètres de la densité conditionnelle de chaque classe $p[I/e_k]$, $k = 1,N$ à partir des prototypes,
2. initialisation :
$$P_i^{(0)}(e_k) = P_i(e_k/I) = \frac{p[I/e_k]P[e_k]}{\sum_{j=1}^{N} p[I/e_j]P[e_j]}$$
La distribution a priori de chaque classe est supposée uniforme,
3. calcul des coefficients de compatibilité sur le voisinage de chaque pixel considéré,
4. calcul du facteur de mise à jour en fonction du coefficient de compatibilité,
5. calcul de $P_i^{(n)}$ à partir de $P_i^{(n-1)}$ et du facteur de mise à jour,
6. recommencer tant que le facteur de mise à jour est supérieur à un seuil.

Nous ne détaillons pas plus le calcul de probabilité d'étiquetage. On attribue à un pixel donné la classe maximisant cette probabilité. Cette méthode comporte quelques désavantages : le nombre de classes doit être fixé a priori. D'autre part, la complexité de calcul est relativement élevée même en ne prenant en compte que les interactions sur un voisinage 8*8. Il est possible tout de même d'accélérer la convergence en décidant qu'une probabilité est nulle lorsque celle-ci est inférieure à un seuil donné. Cette méthode donne de bons résultats pour des images texturées.

1.2.2.2 Classification à partir d'attributs statistiques

On traite ici de la classification à partir d'attributs statistiques provenant soit d'échantillons soit de l'image elle-même. Une texture peut être décrite à partir d'attributs statistiques comme la variance, les paramètes de la matrice de cooccurrence etc.. Ceux-ci sont détaillés en Annexes A. Un vecteur caractéristique d'une texture est composé de plusieurs attributs statistiques quantitatifs. Il est donc de la forme $\mathbf{a} = (\mathbf{a}_1,..,\mathbf{a}_n)^T$, où $\mathbf{a}_i$ est le i ème descripteur de la texture considérée.

Il est possible d'utiliser des techniques d'analyse de données en segmentation de textures ainsi qu'en classification. En effet, cette technique mathématique permet une description du contenu d'un système d'informations afin d'expliquer leurs états en vue d'une action. Les deux centres d'intérêts de l'analyse de données sont la représentation de données et leur classification.

L'analyse factorielle concerne la représentation géométrique dans un espace euclidien de faible dimension d'informations contenues dans un tableau. L'ajustement linéaire est une sous-classe de cette approche. Elle concerne la détermination aux

moindres carrés de la droite ajustant au mieux un nuage de points 2D provenant d'un tableau. On peut distinguer trois types d'analyse factorielle :

- analyse en composante principale (ACP) : représentation des données dans un espace euclidien de dimension faible par projection,
- analyse en correspondances binaires : étude de la relation (dite correspondance) entre deux variables qualitatives,
- analyse en correspondances multiples : généralisation à plusieurs variables.

Nous n'allons pas nous étendre davantage sur le problème de la représentation de données pour nous consacrer à la classification automatique de celles-ci. Les techniques de classification automatique en analyse de données peuvent se décomposer en deux classes :

- les méthodes de partitionnement : elles consistent à déterminer une partition $\mathcal{P}$ d'un ensemble I en regroupant au mieux les éléments de celui-ci,
- les méthodes hiérarchiques : étant donné un ensemble I, la classification est réalisée en trouvant un arbre dont les éléments terminaux sont en bijection avec les éléments de I.

Nous détaillons plusieurs techniques issues de chaque classe par la suite.

a) Les méthodes de partitionnement

Chercher la meilleure partition d'un ensemble I au sens d'un critère consiste à examiner toutes les partitions possibles de I. Or, c'est un problème NP-complet car on peut exprimer le nombre de partitions $Npar$ d'un ensemble à n éléments en k classes de la façon suivante :

$$Npar(n,k) = k * Npar(n-1,k) + Npar(n-1,k-1) \quad k > 1$$

Pour réaliser ces méthodes, plusieurs contraintes sont imposées comme le choix du nombre de classes ou de la taille minimale de celles-ci.

a.1) Maximisation de l'inertie

Etant donnée une partition en k classes d'un nuage de n points dans $\mathbb{R}^p$, on définit les quantités suivantes : $B_1,..,B_k$ les barycentres de chaque classe et $IN_1,..,IN_k$ les inerties des classes. L'inertie est la moyenne des carrés des distances au centre de gravité. L'inertie totale IN_T des n points autour du centre de gravité global B s'écrit sous la forme :

$$IN_T = IN_W + IN_B$$

où $IN_W = \sum p_i I_i$ est l'inertie intraclasse (p_i poids de la classe i) et $IN_B = \sum p_i d^2(B_i,B)$ est l'inertie interclasse ou inertie des k centres de gravité. Le critère de classification détermine la partition telle que IN_B soit maximum.

La connaissance a priori du nombre de classes k est nécessaire.

a.2) Les nuées dynamiques

Les méthodes du type "nuées dynamiques" permettent de traiter rapidement des ensembles d'effectif assez élevé en optimisant localement un critère de type inertie.

Le principe général des ces méthodes consiste à agréger la partition à partir d'éléments appelés "noyaux", puis à partir de la partition obtenue, trouver de nouveaux noyaux capables de générer une meilleure partition. On itère le processus jusqu'à ce qu'il soit stationnaire [DLPT82].

Soit k le nombre de classes fixé a priori, on note $\mathcal{L}_k$ l'ensemble de k noyaux B_i. Les méthodes des nuées dynamiques consistent donc à se donner deux applications f et g. L'application f permet de passer d'une partition à un ensemble de k noyaux et g d'un ensemble de k noyaux à une partition de I. On passe d'une partition $\mathcal{P}_i$ à $\mathcal{P}_{i+1}$ de la façon suivante :

$$\mathcal{P}_{i+1} = [f \circ g]^{i+1}(\mathcal{P}_0)$$

où $\mathcal{P}_0$ est une partition initiale. Ce type d'approche est assez efficace et on montre la convergence de l'algorithme au bout de peu d'itérations.

a.3) La méthode des centres mobiles

Cette technique est une sous-classe des nuées dynamiques où le noyau d'une classe est décrit par son centre de gravité. Le principe de l'algorithme est le suivant :

1. tirage aléatoire de k points B_i de I pour $i = 1..k$,
2. la partition $\mathcal{P}_i$ est obtenue en regroupant les points les plus proches de B_i que tout autre B_j avec $j \neq i$,
3. on remplace les k points pris au hasard par les k centres de gravité de chaque classe ainsi créée,
4. retour en 2 si la condition d'arrêt n'est pas réalisée.

Le critère d'arrêt peut s'exprimer de différentes façons :

- nombre maximal d'itérations fixé,
- aucune modification des centres de gravité.

Cet algorithme nécessite la connaissance du nombre de classes k. La partition en $k+1$ ou $k-1$ classes peut différer assez fortement. Le choix des centres initiaux influe sur la rapidité de convergence de l'algorithme et sur la qualité des résultats. Plusieurs approches ont été proposées en combinant plusieurs itérations de cet algorithme avec des départs initiaux distincts. Quelques améliorations de cet algorithme ont été

suggérées notamment dans la phase du choix des centres initiaux [DLPT82]:

- initialisation par le centre le plus éloigné : On note B_i le i ème centre
  - le centre B_1 est obtenu par tirage aléatoire sur I,
  - le second centre est tel que $B_2 = max_{\{\mathbf{a} \in I\}}\ d(B_1,\mathbf{a})$,
  - pour $j > 2$, $B_j = max_{\{i=1,j-1\}}\ (max_{\{\mathbf{a} \in C_i\}}\ d(B_i,\mathbf{a}))$.
- initialisation par la méthode LBG :
  - on choisit tout d'abord la classe de variance maximale,
  - le centre de cette classe et le point le plus éloigné de celui-ci sont choisis comme points germe de division de la classe,
  - on la divise en deux en appliquant le processus des nuées dynamique.
- initialisation par la méthode du seuillage adaptatif :
  - le centre B_1 est obtenu par tirage aléatoire sur I,
  - chaque vecteur est affecté à ce centre si sa distance est inférieure à un seuil adaptatif en fonction du nombre de centres,
  - le vecteur traité devient un nouveau centre et on recommence jusqu'à obtention du nombre de classes désiré.

Cette dernière méthode est la plus coûteuse en calcul et donne des résultats moyens. Les trois autres méthodes (initialisation aléatoire, centre plus éloigné, LBG) sont relativement équivalentes en complexité mais la méthode LBG est la plus performante.

Il est possible d'améliorer aussi le critère de minimisation en introduisant un terme exprimant la probabilité d'un vecteur d'appartenir à une classe afin d'obtenir une méthode dite floue [Saa96].

a.4) La méthode des plus proches voisins

La méthode du plus proche voisin (1-PPV) consiste à attribuer à un vecteur $\mathbf{a}$ la classe $C = C(\mathbf{a}^{(i)})$ où $i \in \{1,..,NCLASS\}$ désigne l'indice de l'exemple $\mathbf{a}^{(i)}$ le plus proche de $\mathbf{a}$. On formalise cela de la manière suivante :

$$C(\mathbf{a}) = C(\mathbf{a}_i)\ tq\ d(\mathbf{a},\mathbf{a}^{(i)}) < d(\mathbf{a},\mathbf{a}^{(j)}),\ j = 1,NCLASS$$

En cas d'égalité, on pourra effectuer un tirage au sort pour prendre la décision. Il est montré que cette règle converge vers la règle de Bayes mais il n'existe aucun résultat sur la performance pour un nombre fini de vecteurs à classer.

Une généralisation intuitive est la règle des K-plus-proches-voisins (K-PPV). Le procédé est le suivant :

- étant donné un vecteur a à classer, choisir un réel positif r minimum de telle façon que la boule $B(\mathbf{a},r)$ centrée en $\mathbf{a}$ contienne exactement K échantillons,

– affecter **a** à la classe $C(\mathbf{a})$ la plus représentée dans $B(\mathbf{a},r)$.

Pour éviter une prise de décision erronée, on rajoute une clause de rejet pour former une règle (K,K') PPV:

– étant donné un vecteur a à classer, choisir un réel positif r minimum de telle façon que la boule $B(\mathbf{a},r)$ centrée en **a** contienne exactement K échantillons,
– affecter **a** à la classe $C(\mathbf{a})$ la plus représentée dans $B(\mathbf{a},r)$ si ce nombre est supérieur à K' sinon étiqueter **a** par la classe "rejet" signifiant qu'aucune classe n'est satisfaisante.

D'un point de vue pratique, la complexité de l'algorithme est dépendante non seulement de la taille des vecteurs mais aussi du nombre de voisins considérés, à savoir K. Le choix de K est déterminant en ce qui concerne son utilisation : une faible valeur de K permet de distinguer des frontières compliquées entre les classes mais induit une sensibilité au bruit. A contrario, une valeur élevée de K a pour conséquence un lissage des frontières mais une bonne robustesse vis-à-vis du bruit. Il est donc nécessaire de trouver le bon compromis entre performance, robustesse et coût de calcul.

b) Les méthodes hiérarchiques

La détermination d'une classification hiérarchique peut être réalisée de deux façons [DLPT82] : la première méthode consiste à regrouper les éléments très semblables dans un premier temps puis, à partir de ces classes, on en construit d'autres de moins en moins homogènes. Cette approche s'appelle la classification hiérarchique ascendante. La seconde technique effectue une dichotomie successive de l'ensemble à classer; ce mode de construction se nomme la classification hiérarchique descendante.

Soit $\mathcal{O}$ l'ensemble des individus à classer, une famille H de parties de $\mathcal{O}$ est une hiérarchie si :

1. $\forall a \in \mathcal{O},\ \{a\} \in H$ et $I \subseteq H$
2. $\forall A,B \in H,\ A \cap B \in \{A,B,\emptyset\}$
3. $\forall A \in H,\ \exists\{A_i,\ i=1,p\}\ tq\ A = \cup_i A_i\ et\ A_i \subset A$

La propriété 2 signifie que deux classes sont soit disjointes, soit contenues l'une dans l'autre. La dernière condition affirme qu'à toute hiérarchie correspond un arbre de classification. On illustre ceci par la figure 1.1 représentant l'arbre de classification associé à une hiérarchie donnée.

En pratique, l'agrégation ascendante est utilisée dans la plupart des cas. On considère donc un tableau de données assimilable à un tableau de coordonnées individus-variables. La construction d'une hiérarchie nécessite la connaissance d'une mesure de ressemblance entre éléments et entre sous-ensembles de $\mathcal{O}$. Cette mesure est appelée indice d'agrégation. On note $\mathcal{O}_1$ et $\mathcal{O}_2$ deux sous-ensembles de $\mathcal{O}$, nous présentons

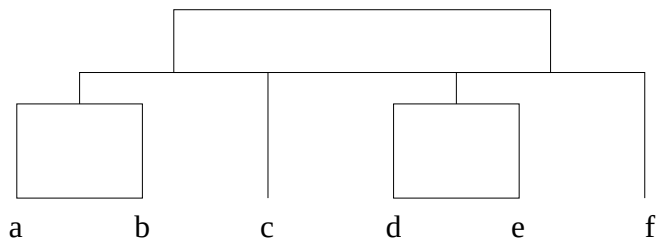


FIG. 1.1 – *Arbre de classification correspondant à la hiérarchie suivante* $H = \{\emptyset, a, b, c, d, e, f, ab, abc, de, def, abcdef\}$

une liste de quelques indices d'agrégations existants [DLPT82]:

– indice d'agrégation du lien minimum :

$$d_a(\mathcal{O}_1,\mathcal{O}_2) = min_{o_1 \in \mathcal{O}_1, o_2 \in \mathcal{O}_2} d(o_1,o_2)$$

– indice d'agrégation des barycentres :

$$d_a(\mathcal{O}_1,\mathcal{O}_2) = d(B(\mathcal{O}_1),B(\mathcal{O}_2))$$

– indice d'agrégation des moyennes des distances :

$$d_a(\mathcal{O}_1,\mathcal{O}_2) = \frac{1}{card(\mathcal{O}_1)card(\mathcal{O}_2)} \sum_{o_1 \in \mathcal{O}_1, o_2 \in \mathcal{O}_2} d(o_1,o_2)$$

On réunit à partir de cette distance les éléments les plus proches puis on réitère pour n'obtenir qu'une seule classe s'écrivant comme la réunion de toutes les autres.

Le choix de la distance entre deux sous-arbres est importante et nécessite une étude approfondie. Une limitation de l'utilisation de ces méthodes concerne le coût important de calcul (calcul de la dissimilarité en $O(\frac{mn(n-1)}{2})$ où m est le nombre de variables et n celui des individus).

Des techniques de classification hiérarchiques plus évoluées existent néanmoins permettant de pallier l'inconvénient du coût de calcul. On peut citer la méthode CHAVL (Classification Hiérarchique par Analyse de la Vraisemblance des Liens) permettant de traiter des variables quantitatives mais aussi qualitatives avec efficacité rapidement. L'originalité de l'approche repose sur la définition d'indices probabilistes de similarité entre objets élémentaires [LPL93]. La notion de vraisemblance de liens entre deux classes est considérée dans l'indice d'agrégation utilisé. Il faut noter que cet indice diffère pour classer des variables et des individus.

Dans le cas de classification de variables, le critère d'agrégation est le suivant :

$$VL_\epsilon(\mathcal{A}_1,\mathcal{A}_2) = [p(\mathcal{A}_1,\mathcal{A}_2)]^{(card(\mathcal{A}_1)card(\mathcal{A}_2))^\epsilon}$$

où $p(\mathcal{A}_1,\mathcal{A}_2) = max\{P(a_1,a_2)/(a_1,a_2) \in \mathcal{A}_1 \times \mathcal{A}_2\}$. Le terme $P(a_1,a_2)$ est l'indice de vraisemblance de lien [LP85].

Outre le caractère générique de la méthode de classification (possibilité de considérer des variables qualitatives), l'originalité de la méthode réside dans la différence de stratégie adoptée de classification des variables et d'individus. Ainsi, la similarité entre deux individus est calculée à partir de la contribution de chaque variable à les différencier.

1.2.3 Discussion

Les méthodes de détection de contours sont efficaces pour des images faiblement bruitées ou texturées. Il faut noter cependant la difficulté d'ajustement des paramètres de ce type d'approche (seuils). D'autre part, ces méthodes engendrent des sur-segmentations même dans le cas d'images faiblement bruitées.

Les méthodes de croissance de régions ont le même domaine d'application. Néanmoins, la localisation des frontières entre régions est beaucoup moins précise et les conditions d'arrêt de la croissance sont difficiles à fixer.

Le tableau 1.1 récapitule les avantages et inconvénients de chaque type de méthodes adaptées aux images uniformes.

Type de méthode	**Connaissances**	**Mode non supervisée**	**Rapidité**
Détection de contours	seuils	non	bonne
Croissance de régions	critères d'arrêt	oui	bonne

TAB. 1.1 – *Avantages et inconvènients des méthodes de segmentation adaptées aux images uniformes*

Les méthodes bayésiennes sont adaptées aux textures aléatoires microscopiques. Elles présentent néanmoins le désavantage de faire des hypothèses sur la distribution des observations et des étiquettes. Lorsque ces hypothèses ne sont pas vérifiées a posteriori, il peut en résulter des résultats de segmentation de mauvaise qualité.

Les méthodes de classification à partir d'attributs sont efficaces pour partitionner un ensemble de points par apprentissage ou non. Le point essentiel de ce type de méthode est le choix des attributs utilisés qui n'est pas toujours en adéquation avec l'image à traiter. Il faut noter en outre que peu d'approches de ce type permettent d'obtenir le nombre de classes présentes dans l'image.

Le tableau 1.2 récapitule les avantages et les inconvénients de deux types de méthodes adaptées aux images texturées.

Type de méthode	Connaissances	Mode non supervisée	Rapidité
Méthode bayésienne	hypothèses sur distributions	oui	moyenne
Classification à partir d'attributs	taille d'analyse	oui	bonne

TAB. 1.2 – *Avantages et inconvénients de deux types de méthodes adaptées à la segmentation d'images texturées*

1.3 Méthodes de segmentation coopératives

On se rend compte assez vite de la limite des méthodes détaillées auparavant. En effet, la plupart de ces méthodes sont efficaces exclusivement pour un certain type d'images. Quelques études se sont focalisées sur la coopération de plusieurs méthodes de segmentation dont le but n'était pas forcément de généraliser mais de combiner les avantages des méthodes utilisées. On retrouve dans cette optique trois approches de coopération :

- les méthodes séquentielles,
- les méthodes parallèles,
- les méthodes hybrides.

Nous détaillons quelques méthodes issues de chaque classe.

1.3.1 Méthodes séquentielles

Les méthodes de coopération séquentielle combinent différentes méthodes afin d'obtenir le résultat de sgmentation final. Chronologiquement, ce type d'approche est apparu afin de combiner les méthodes de détection de contours et de croissance de régions.

a) Coopération contour-régions

L'application d'une technique de croissance de régions peut introduire plusieurs erreurs. En effet, intuitivement un contour dans l'image segmentée doit correspondre à une frontière entre deux régions ou à un contour dans l'image originale. Or la croissance de régions ne garantit pas cette propriété. L'idée a donc été d'introduire un processus de détection du contour à l'intérieur de la croissance de régions. Les contours sont en général utilisés comme des contraintes, ou des références, permettant de guider et de contrôler les processus d'extraction des régions.

Wrobel et Monga [WM87] proposent une technique de segmentation régions guidée par des points de contour obtenus par extraction préalable. Leur algorithme fonctionne en trois étapes. La première consiste à réaliser une segmentation à l'aide d'une procédure de type fusion de régions avec le critère $Min - Max$. Le résultat de cette segmentation est ensuite représenté sous forme d'un graphe d'adjacence lors de la seconde phase. Chaque noeud du graphe est associé à une région et les arcs représentent les relations d'adjacence entre ces régions. La dernière étape consiste

à regrouper les couples de régions voisines sur la base d'un critère dépendant de la variance de la région totale et du pourcentage de points contour sur la frontière commune aux deux régions à fusionner.

Une autre approche développée par Bhalero [BW91] utilise conjointement des informations régions et contours à partir de graphes d'adjacence. Elle adopte une approche multirésolution pour assurer la robustesse vis-à-vis du bruit. Les étapes de son algorithme sont les suivantes :

- lissage de l'image par une pyramide passe bas,
- création d'un graphe d'adjacence des régions à partir de la pyramide créée précédemment,
- création d'un graphe d'adjacence des contours à partir de la pyramide,
- fusion des informations contenues dans chaque graphe.

b) Coopération multi-seuillage et fusion de régions

Cette méthode développée au LASTI [KCC95] est basée sur la combinaison de deux processus de segmentation. Cette méthode automatique utilise, dans une première étape, une classification à partir des histogrammes locaux. La seconde étape est une fusion de régions étiquetées minimisant un critère de similarité. Les différentes étapes de cet algorithme sont :

- calcul d'histogrammes locaux,
- transformations des histogrammes : les histogrammes locaux sont transformés en utilisant un critère d'homogénéisation prenant en compte les 4 voisins les plus proches. Un lissage par une gaussienne est réalisé dans une seconde étape. Cette étape met en évidence les pics les plus significatifs des histogrammes,
- recherche des pics significatifs : un pic est dit significatif s'il est au moins séparé d'une distance minimale des autres pics significatifs et sa hauteur doit être supérieure à une fraction du maximum de l'histogramme. Ces seuils ne sont pas fixés empiriquement mais s'adaptent en fonction de l'histogramme traité,
- pré-segmentation de l'image par multi-seuillage, on obtient une sur-segmentation de celle-ci,
- fusion des régions ayant un indice de similarité proche.

Cette méthode est totalement non supervisée et donne de bons résultats pour des images texturées.

D'autres méthodes séquentielles ont été mises au point, elles utilisent une approche multi-résolution afin d'appliquer une méthode de segmentation sur l'image à différentes résolutions. Ce type d'approche utilisé [BKS98], [BS94], [Bur84] présente l'avantage d'être robuste au bruit et permet d'accélérer considérablement les traitements.

1.3.2 Méthodes parallèles

Ces méthodes tentent de combiner ou de fusionner plusieurs résultats de segmentation de la même image en utilisant plusieurs méthodes.

1.3.2.1 Coopération contour-région

Farrokhinia et al [FJ91] ont proposé une méthode combinant une détection de contours et une classification de pixels par CLUSTER. La méthode est la suivante :

- calcul d'images attributs à partir des filtres de Gabor,
- classification des pixels de l'image avec CLUSTER (moindres carrés),
- utilisation d'un filtre adaptatif de Canny afin de détecter les contours sur les images attributs,
- intégration du résultat de chaque segmentation en ne conservant que les contours significatifs c'est-à-dire ceux correspondant à des frontières entre deux textures.

Cette méthode est assez coûteuse tant par la dimension élevée de l'espace d'attributs mais aussi par l'utilisation de deux techniques de segmentation et de l'étape nécessaire d'intégration de résultats. Cette solution a néanmoins l'avantage d'être particulièrement peu sensible au bruit.

1.3.2.2 Fusion de résultats de segmentation

Beaucoup de méthodes de coopération parallèle se réduisent à une fusion de plusieurs résultats de segmentation d'une même image. L'approche conjonctive (correspondant à un ET logique) permet de fusionner des informations en ayant un jugement sévère ou prudent car seule la partie commune est conservée [Blo96]. L'approche disjonctive correspond à un mode de raisonnement souple ou indulgent ce qui est justifié lorsque les sources d'informations sont indépendantes ou si l'une d'entre elles est supposée érronée.

La théorie des ensembles flous introduite par Zadeh [Zad65] consiste à quantifier un ensemble par un degré d'appartenance s'interprétant comme une probablilité de l'événement correspondant. L'inconvénient de cette approche est le choix des fonctions d'appartenance souvent obtenues par apprentissage [Blo96]. La théorie des possibilités est dérivée des ensembles flous et permet de définir la notion d'incertitude.

La théorie de Dempster-Schafer [DLR77] est un formalisme utilisant la notion de croyance d'un événement. Cette approche est assez utilisée en fusion de résultats de segmentation d'images [DKZB99]. La théorie des probabilités reste néanmoins la plus utilisée [Laf96]. Elle repose sur la théorie bayésienne de l'information et est considérée comme étant la plus performante [Blo96].

Récemment, une méthode générale d'intégration de résultats de plusieurs segmentations a été suggérée en utilisant le principe de segmentation floue [fB95]. Le principe de la méthode est le suivant :

1. on dispose de n résultats de segmentations (issus de la même technique ou non) et on crée une région floue à partir des n régions nettes précédemment obtenues de la façon suivante. Soient R_1,..,R_n les représentations d'une région R de l'image, on définit les ensembles suivants :

   $N_n(R) = \cap_i R_i$: ensemble des pixels appartenant certainement à R,
   $S_n(R) = \cup_i R_i$: ensemble des pixels ayant une possibilité d'appartenir à R.

   On pose $confiance(R_i)$ le degré de confiance en le résultat de la i ème segmentation de la région R avec la propriété $\sum_i confiance(R_i) = 1$. Si on ne dispose pas d'informations a priori sur la qualité de chaque segmentation, on la suppose uniforme. Une région floue R est définie à partir des représentations R_1,..,R_n et d'une fonction d'appartenance s'écrivant sous la forme :

   $$\forall s \in I,\ \mu_R(s) = \sum_{j/s \in R_j} confiance(R_j)$$

   Plus cette fonction est faible pour un pixel donné, moins il y a de chances qu'il appartienne à la région R.
2. cette étape consiste à mettre en correspondance des régions issues de différentes segmentations : supposons avoir réalisé n segmentations où $R_i = (R_i^1,..,R_i^{nb(i)})$ est l'ensemble des $nb(i)$ régions détectées lors de la i ème segmentation. On crée une liste $\{\tilde{R}(1),..,\tilde{R}(c)\}$ supposée être l'ensemble des véritables régions de l'image où chaque région $\tilde{R}(i)$ est représentée par l'ensemble $\{\tilde{R}^1(i),..,\tilde{R}^n(i)\}$. Une région $\tilde{R}^j(i)$ est associée à la région $\tilde{R}(i)$ lors de la j ème segmentation.
3. on définit chacune de ces régions supposées véritables comme des régions floues avec comme fonction d'appartenance pour la région $\tilde{R}(i)$:

   $$\forall s_k \in I,\ \mu_{ik}(x) = \sum_{j/s_k \in \tilde{R}(i)} confiance(R_j)$$

   où $confiance(R_j)$ est le degré de confiance en la segmentation R_j et μ_{ik} est le degré d'appartenance du pixel s_k à la région $\tilde{R}(i)$. On construit ainsi une segmentation floue avec la matrice d'appartenance :

   $$\Gamma = (\mu_{ik},\ 1 \leq i \leq c,\ 1 \leq k \leq nm) \quad avec \quad \sum_{i=1}^{c} \mu_{ik} = 1,\ 1 \leq k \leq N$$

4. la dernière étape consiste à passer à une segmentation nette à partir de cette segmentation floue. On affecte le pixel s_k à la région $\tilde{R}(j)$ si $\mu_{jk} = max_i(\mu_{ik})$

pour $1 \leq i \leq c$. On opère une relaxation sur l'étiquetage par l'algorithme décrit auparavant pour améliorer la qualité des résultats.

1.3.3 Méthodes hybrides

Les méthodes coopératives dites hybrides utilisent des concepts plus élaborés telles que la remise en cause de résultats intermédiaires, l'adaptation au contexte et en fonction des résultats obtenus.

1.3.3.1 Coopération active

L'approche proposée par Bajcsy [BAJ88] exploite la complémentarité région-contour et consiste à faire coopérer deux algorithmes de segmentation alternativement en boucle fermée, jusqu'à ce que les résultats obtenus soient jugés satisfaisants. Le système est composé de deux modules : un détecteur de contours et un détecteur de régions. Plusieurs paramètres de contrôle sont fournis en entrée de chaque module afin de régler la formation des contours et des régions. En sortie, chaque module comporte des paramètres qui permettent de quantifier les résultats obtenus. Le résultat final de la segmentation est obtenu après exécution d'un certain nombre de boucles sur quelques ou sur tous les modules en même temps.

Le processus est itératif, les résultats de la segmentation obtenus lors d'une itération (quantifiés par les paramètres de sortie) sont utilisés pour adapter au mieux le ou les modules lors de l'itération suivante. Chaque module peut fonctionner en boucle fermée sur lui-même en ajustant ses propres paramètres d'entrée en fonction de ses sorties. Dans le cas où le système global fonctionne en boucle fermée, c'est-à-dire les deux modules chaînés, son résultat (image de régions) adapte les paramètres d'entrée de chaque module.

L'algorithme de détection de contour utilisé est celui de Canny [Can86], qui présente l'avantage d'être paramétrable. La détection de région s'opère par un algorithme de croissance de régions. Le module de détection de contours utilise deux paramètres en entrée : la taille de la fenêtre de l'opérateur de lissage utilisée par le filtre de Canny et le seuil sur la norme du gradient. En sortie, le détecteur de contours mesure deux paramètres : la densité de points de contours de l'image et la longueur moyennes des contours.

Pour le module d'extraction de régions, les paramètres d'entrée sont : le seuil sur la différence locale admise à l'intérieur d'une même région, la taille minimale de la région, et un paramètre de convexité. En sortie, deux attributs sont calculés : la taille moyenne de régions obtenues (ou le nombre) et le coefficient de compacité moyen de régions (donné par le rapport périmètre/aire).

Le paramètre de retour le plus important est le nombre de régions. En effet, le nombre de régions désirées est une donnée initiale. L'algorithme va modifier automatiquement ses paramètres de contrôle en entrée, à l'aide de rebouclages, de façon à ajuster au mieux le nombre de régions trouvées à l'issue du processus de segmentation. Cette procédure est ainsi itérée tant que le nombre de régions trouvées n'est pas similaire à celui désiré. En fait, tous les paramètres du système sont liés. Le nombre de régions dans l'image permet de déduire leur taille moyenne. L'estimation du rapport moyen entre le périmètre et l'aire des régions permet de donner une indication sur le facteur d'échelle du détecteur de contour, ainsi que la densité de points de contours dans l'image. Ce paramètre permet moyennant la boucle de rétroaction sur le détecteur de contours, de fixer la valeur du seuil sur la norme du gradient. Ce dernier paramètre est lié au seuil sur la différence locale des niveaux de gris admise à l'intérieur d'une région. La taille minimale des régions est estimée par un pourcentage de la taille moyenne des régions désirées.

Cette méthode constitue une approche intéressante pour résoudre le problème de segmentation, elle présente toutefois les inconvénients suivants :

- d'une part, l'introduction de connaissances a priori trop précises (le nombre de régions désirées et le rapport périmètre sur aire). En général, cette information n'est pas disponible,
- d'autre part, les règles précises de contrôle de la convergence du processus ne sont pas présentées.

Plusieurs systèmes de segmentation ont tenté de combiner l'approche contour permettant une bonne localisation des frontières et l'approche région plus robuste vis à vis du bruit [FBC94], [CA93]. Le principe consiste généralement à contrôler la croissance de régions par les contours. Ces méthodes ne donnent pas de bons résultats dans le cas texturé puisque la détection de contours n'est pas une information pertinente dans ce cas. Afin de pallier cet inconvénient, un système intégrant la texture a été proposé récemment [KC98].

Ce système est composé de deux modules. Le premier est dédié à l'extraction de régions uniformes ou faiblement texturées. Le principe utilisé est une coopération contour-région jusqu'à convergence d'une mesure de cohérence. Le second module localise les régions texturées et contribue à la réactualisation et à la correction des contours extraits par le premier module.

Cette approche se distingue des précédentes dans la mesure où la segmentation est adaptative *a posteriori*, c'est à dire qu'elle met en évidence les zones de l'image non correctement traitées et corrige ces erreurs en relâchant des contraintes sur les méthodes de segmentation employées.

1.3.3.2 Coopération multi-agents

L'approche multi-agents pour effectuer une coopération région-contour semble naturelle (chaque agent est un petit système expert autonome lié aux autres agents). Baujard et Garbay [BG91] proposent un système de vision complet pour l'identification de cellules biologiques, fondé sur la coopération entre le détecteur de contours de Deriche et la technique de classification de Fisher appliquée aux régions, précédée d'un filtrage de l'image. Un agent est dédié à chacun des processus. Après détection des régions et des contours, les résultats sont analysés et comparés par un agent spécifique. Lorsqu'il y a compatibilité entre la frontière des régions et les contours, le résultat est validé et envoyé à un agent qui tente de classer la région à partir de la forme de celle-ci. Si la compatibilité n'est pas vérifiée, les régions sont découpées ou fusionnées.

Cette approche est particulièrement intéressante car elle pose clairement le problème de la gestion de la coopération. Il ne s'agit pas ici de juger cette méthode sur la qualité des résultats mais sur la manière d'aborder le problème. Il semble en effet qu'il y ait de grandes difficultés à faire coopérer deux méthodes qui ont peu de points communs et qui ne sont pas spécialement préparées pour cette coopération. Si le cumul des informations est important, la complémentarité des informations doit également être prise en compte pour que l'efficacité soit réelle. Ainsi, à partir d'un détecteur de contours, il est difficile de trouver un processus de détection des régions complémentaire.

La distribution des rôles dans un univers multi-agents n'est donc pas remise en cause, mais elle n'est pas très efficace si les processus coopérants sont trop indépendants. Cette remarque reste valable pour les coopérations non fondées sur l'approche multi-agents. Il est difficile de cerner les avantages théoriques de l'approche multi-agents pour la coopération région-contours. Si chaque agent est distribué sur un processeur différent, il y a bien sûr un gain de temps. En ce qui concerne la qualité de la segmentation, il nous semble que les informations sur les régions et les contours doivent être exploités simultanément au niveau local. Peut-être faudrait-il donc distribuer spatialement les agents. De manière générale, l'approche multi-agents et les algorithmes parallèles méritent une plus grande attention car la façon d'aborder les problèmes incite à une réflexion profonde sur la gestion des informations et l'enchaînement des étapes.

Une autre illustration de ce type de méthode est donnée dans [SGC95]. Dans ce cas, plusieurs agents coopèrent de manière interactive. Il y a deux types d'estimation, l'un concernant le bruit et l'autre des paramètres de la texture dans l'image. En fonction de cette estimation, une détection de contours locale est réalisée. Cette coopération s'appuie surtout sur une architecture dédiée permettant le choix de la méthode appropriée compte tenu du contexte. D'autres travaux existent par ailleurs

et dont le détail est donné dans [BB92] et [Bon91].

Une approche système a été proposée récemment par F. Bellet [Bel98] adoptant une approche incrémentale. En plus de sa complexité algorithmique, cette approche se révèle être assez lourde en temps de calcul et ne fournit pas toujours une segmentation complète de l'image.

1.3.3.3 Coopération adaptative

Une méthode coopérative supervisée a été proposée par Salzenstein [SP95]. Le choix de la méthode la plus adaptée est conditionné en fonction des caractéristiques de l'image en entrée mais aussi dépend du taux d'erreur de classification. La vérité-terrain de l'image est donc nécessaire, cette méthode a été testée avec des images de synthèse perturbées par du bruit. Trois méthodes statistiques ont été choisies pour la coopération :

- approche globale champ de Markov : estimation des paramètres par ICE et segmentation par MPM,
- approche locale : ICE aveugle adaptative non stationnaire,
- chaînes de Markov cachées : estimation des paramètres par EM et segmentation par MPM.

Plusieurs critères d'évaluation ont été choisis pour évaluer la qualité de la segmentation :

- trois critères en amont :
  - homogénéité,
  - type de bruit,
  - corrélation spatiale.
- un critère en aval : une méthode est dite meilleure qu'une autre si son taux d'erreur de classification est inférieur de 4%.

Une étude a été réalisée pour évaluer la performance de chaque méthode en faisant varier la valeur des 3 critères en amont. La segmentation permet alors de choisir la méthode appropriée en fonction des trois critères cités précédemment et du résultat de l'étude d'évaluation. L'avantage de cette méthode est qu'il n'est pas nécessaire de segmenter l'image par les trois méthodes. Par contre, le principal inconvénient réside dans la nécessité d'évaluer les performances de chaque méthode en faisant une étude préliminaire dépendant des critères choisis en amont.

Un système à base de règles a été proposée par Nazif et Levine [LN85]. Le système se focalise sur des régions de types différents : région uniforme, texturée et contour. Cette distinction entre ces trois types de régions est obtenue à partir d'une pré-segmentation d'une image et d'une image contour obtenue par un gradient par exemple. Cette analyse de l'image permet d'adapter les traitements par la suite.

Même si la méthodologie adoptée est intéressante, elle reste néanmoins assez limitée (2 algorithmes de segmentation) et s'appuie sur une analyse moins fine de l'image.

1.3.4 Discussion

De plus en plus de travaux portent sur la coopération de méthodes de segmentation. Des techniques prometteuses existent sans toutefois exploiter tous les concepts existants. Il faut en effet mettre l'accent sur quelques difficultés limitant l'application de tels concepts.

L'approche séquentielle ne peut donner de bons résultats que lorsque toutes les méthodes dans la chaîne de traitement produisent un bon résultat intermédiaire. Or, il n'existe pas de méthode génerale permettant de traiter tout type d'images avec efficacité.

L'approche parallèle permet effectivement de s'affranchir de cet inconvénient majeur. Par contre, la conception de tels systèmes est délicate dans la mesure où à la fois la complémentarité et la redondance d'informations issues de chaque résultat doivent être prises en compte. De plus, une information redondante n'est pas forcément pertinente. En effet, si les méthodes utilisées ne sont pas adaptées au contexte, la combinaison de leurs résultats sera de mauvaise qualité.

L'approche hybride est plus flexible puisqu'une méthode défaillante pour une image n'entrainera pas forcément un résultat de segmentation erroné. La remise en cause d'un résultat intermédiaire peut se révéler être un atout majeur dans un système de segmentation. Cela nécessite l'élaboration de critères d'évaluation de résultats intermédiaires.

1.4 Conclusions

Cette synthèse bibliographique nous a permis de tirer plusieurs conclusions.

Tout d'abord, nous avons montré que la plupart des méthodes de segmentation non coopératives donnent de bons (voire très bons) résultats lorsque l'image à traiter entre dans le domaine d'applications pour lesquel elles ont été développées. Ceci montre que pour un type d'images particulier, il est possible de trouver une méthode de segmentation appropriée.

Ces méthodes adoptent une même stratégie de traitement pour toute l'image. Pourtant, les images réelles sont rarement complètement uniformes ou texturées. Ceci montre l'intérêt d'adopter des stratégies différentes suivant le contexte local.

Les techniques coopératives sont apparues d'une part, pour combiner les avantages des méthodes précédentes mais aussi d'autre part, pour permettre de traiter localement l'image à segmenter. Comme nous l'avons vu, la plupart de ces méthodes tentent de segmenter en aveugle l'image sans considérations sur le contexte de l'image. Cette méthodologie couramment adoptée en segmentation d'images paraît relativement surprenante. En effet, dans la mesure où la plupart des méthodes de segmentation font implicitement des hypothèses sur l'image en entrée (image uniforme, texturée, faiblement bruitée,..), pourquoi ne pas essayer de vérifier ces hypothèses avant d'utiliser une méthode particulière ? L'approche actuelle consiste plutôt à identifier l'échec d'une méthode (par exemple sur-segmentation de zones texturées) et à exploiter cette information pour déclencher un processus plus complexe. S'il est vrai que parfois les hypothèses sont difficiles à vérifier à partir de l'image, ce problème n'est pas toujours insurmontable. En effet, il existe des critères très simples permettant par exemple de différencier une région texturée d'une uniforme.

Pour aboutir à une segmentation fiable d'une image, celle-ci doit être réalisée en appliquant localement la méthode de segmentation la plus adaptée aux données. Lors de la conception d'un système de segmentation, l'effort doit porter sur l'analyse locale de l'image et sur la mise en correspondance d'un contexte local et d'une ou plusieurs méthodes de segmentation appropriée. Nous appelons par la suite ce type d'approche la segmentation adaptative.

Notre travail s'incrit dans cette démarche de conception de sytèmes de segmentation. En effet, nous avons choisi d'adapter la segmentation d'une image en considérant dans un premier temps, le type de chaque région la composant : région texturée ou uniforme. Comme nous l'avons vu, ce choix semble justifié car les méthodes adaptées à ces types de région sont relativement distinctes. Nous proposons d'affiner, dans un second temps, la caractérisation des régions texturées en considérant le type de texture présent (texture aléatoire, déterministe, grossière, fine). La segmentation de textures étant beaucoup plus délicate, ce type d'information est nécessaire pour mieux adapter les traitements.

La stratégie globale du système étant définie, la difficulté réside dans la sélection des méthodes de segmentation appropriée à chaque type de rgion. Dans la mesure où nous portons un intérêt tout particulier à l'aspect automatique et non supervisée du dispositif, nous avons choisi une méthode de segmentation générale et non supervisée basée sur la classification automatique à partir d'attributs de l'image. Ce type d'approche présente de nombreux avantages :

- efficacité en segmentation d'images uniformes et texturées,
- adaptation naturelle au contexte en choisissant les attributs appropriés à chaque type de régions (texturée, uniforme) et la taille de la fenêtre d'analyse,

– détermination automatique possible du nombre de classes de l'image et convergence prouvée mathématiquement.

PARTIE II

Système adaptatif de segmentation d'images

- **Introduction**

Les efforts de recherche dans le domaine de la segmentation d'images ont conduit à la réalisation d'une quantité impressionnante d'algorithmes aussi divers que les applications abordées. Néanmoins, beaucoup de systèmes restent encore trop dépendants de leur application pour pouvoir être réutilisables facilement.

Pour une plus grande efficacité de ces systèmes, il est nécessaire qu'ils possèdent des capacités de généricité, de flexibilité et d'adaptabilité à la variabilité des tâches et des contextes. Il faut donc concevoir des systèmes qui puissent intégrer sous différentes formes les connaissances et les informations disponibles, pour leur permettre d'identifier dynamiquement le problème présenté et lui apporter la solution la plus appropriée.

La résolution des problèmes de traitement d'images nécessite tout d'abord l'introduction de critères décisionnels permettant à partir de données d'entrées abondantes, incomplètes ou imprécises, de choisir les algorithmes les plus appropriés puis, si nécessaire, d'évaluer les résultats pour savoir quand le problème peut être considéré comme résolu.

Nous proposons dans le cadre de cette thèse un système de segmentation permettant d'adapter localement la segmentation d'une image étant donné son contexte. On peut en effet espérer une meilleure efficacité en adaptant un traitement aux données plutôt qu'en appliquant un seul outil général. Dans le cas où plusieurs méthodes sont adaptées à un type d'image, les résultats de ces méthodes sont intégrés dans le résultat final de segmentation. Pour pouvoir appliquer ce système à des images multi-composantes, une méthode de fusion adoptant une approche génétique a été développée.

Nous décrivons, dans un premier temps, le principe de fonctionnement du système de segmentation que nous proposons. Nous détaillons, dans un deuxième temps, ses différents modules. Nous concluons et donnons quelques perspectives au système développé.

- **Principe du système de segmentation proposé**

Afin de concevoir un système de segmentation, il est primordial d'adapter la segmentation suivant le contexte. Cette approche s'inspire de la vision humaine. En effet, lorsque nous analysons une scène, notre oeil balaie les différents centres d'intérêt de celle-ci à l'aide de la rétine fovéale tandis qu'une perception globale floue est obtenue par la rétine périphérique. Chaque composante de notre système visuel a une tâche précise à réaliser et coopère afin d'obtenir une perception globale de la scène.

Le système que nous proposons est composé de trois modules : l'analyse de l'image, la segmentation adaptative et la fusion de résultats de segmentation.

- **l'analyse de l'image** : ce module permet d'extraire, dans un premier temps, des paramètres permettant de répertorier les zones de l'image de façon grossière afin d'identifier le contexte de celle-ci (image composée majoritairement de régions texturées ou uniformes). Une région est dite uniforme si la variance de ces niveaux de gris est faible. Elle regroupe donc les zones faiblement bruitées ou faiblement texturées. Cette analyse a pour but d'adapter les traitements postérieurs et de s'affranchir d'éventuelles heuristiques en affectant les pixels d'une image dans l'une des deux classes suivantes :
  - pixels d'une région fortement texturée,
  - pixels appartenant à une zone uniforme ou faiblement texturée.

  Cette partition des régions de l'image est primordiale dans la mesure où les méthodes de segmentation adaptées à chaque type de zone sont relativement distinctes comme nous l'avons vu dans l'état de l'art (cf. chapitre 1). L'affectation d'un pixel dans l'une des deux classes est réalisée en considérant un paramètre local extrait de la matrice de cooccurrence. Afin de rendre l'étape de segmentation adaptative, nous analysons dans un deuxième temps chaque type de région. Pour les régions uniformes, deux attributs sont extraits afin de les décrire. Quant aux régions texturées, nous les avons caractérisées par des attributs classiques et de nouveaux attributs mettant en évidence le caractère aléatoire ou déterministe et la granularité d'une texture,
- **la segmentation adaptative de chaque zone de l'image** : chaque zone précédemment détectée est segmentée en appliquant une ou plusieurs méthodes appropriées. Dans le cas où plusieurs méthodes sont adaptées pour traiter un type de région de l'image, les différents résultats de segmentation obtenus sont intégrés dans le résultat final de segmentation,
- **la fusion de résultats de segmentation** : ce module permet de fusionner plusieurs résultats de segmentation d'une même image ou le résultat de segmentation de chaque bande d'une image multi-composantes. Pour fusionner plusieurs résultats de segmentation, nous avons développé une méthode fondée sur une approche génétique. Cette méthode tient compte de la qualité de chaque résultat de segmentation quantifiée par un critère d'évaluation.

La figure FIG. 1.2 résume le principe générale du système proposé. Nous détaillons à présent les trois modules de ce système dans trois chapitres.

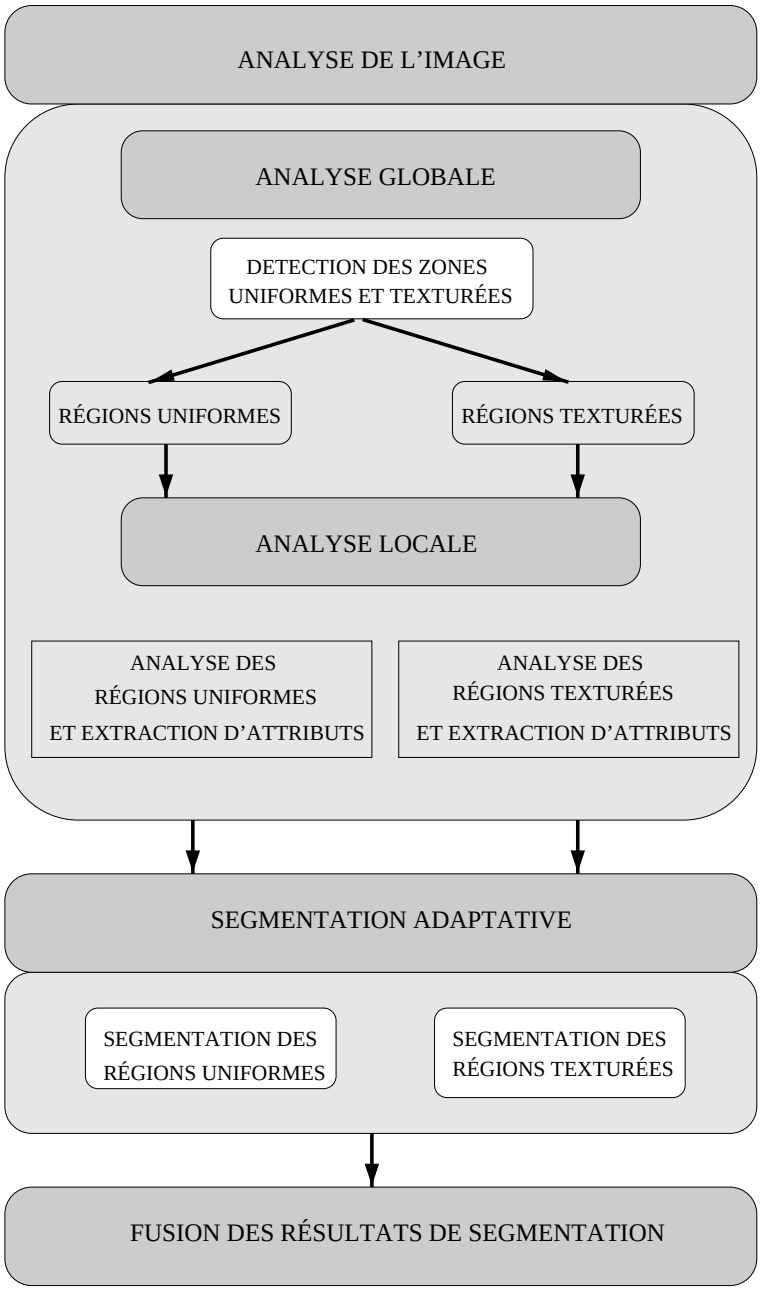


FIG. 1.2 – *Schéma du système adaptatif de segmentation*

Chapitre 2

Analyse d'images

2.1 Introduction

L'analyse d'images est une étape primordiale dans la conception d'un système adaptatif de segmentation. La qualité de la caractérisation des zones d'une image permettra de déterminer avec précision la méthode de segmentation la plus appropriée et ainsi d'obtenir le résultat optimal.

Cette analyse des régions de l'image à segmenter doit permettre la mise en correspondance d'un type de zone détecté et une ou plusieurs méthodes de segmentation. Comme nous l'avons vu dans le chapitre 1, il est possible d'établir une taxonomie des méthodes non coopératives en considérant le type d'images ou de régions pour lesquelles elles sont adaptées. Nous proposons alors de localiser dans un premier temps les régions texturées d'une image et celles uniformes ou faiblement texturées. Cette distinction entre ces deux types de zones pemettra d'appliquer le type de méthode de segmentation approprié. Dans un second temps, nous affinons la caractérisation des zones texturées afin de faciliter leurs segmentations.

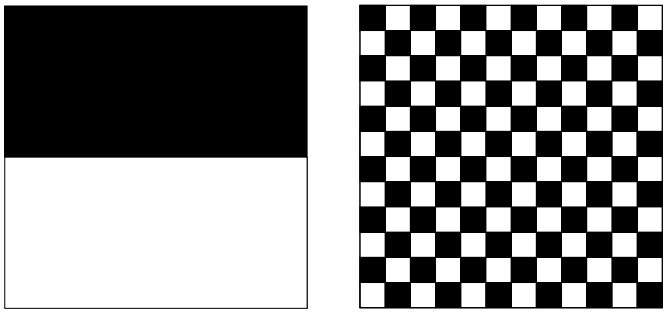
Le module d'analyse d'images proposé se décompose alors en deux sous-modules :

- **l'analyse globale de l'image** : elle permet de répertorier les zones de l'image de façon grossière. Cette analyse a pour but d'adapter les traitements postérieurs et de s'affranchir d'éventuelles heuristiques en affectant les pixels d'une image dans l'une des deux classes suivantes :
  - pixels d'une région fortement texturée,
  - pixels appartenant à une zone uniforme ou faiblement texturée.

  Pour affecter un pixel à l'une des deux classes, nous avons utilisé un paramètre d'homogénéité calculé à partir de la matrice de cooccurrence,
- **l'analyse locale de l'image** : elle permet de caractériser chaque type de région par un ensemble d'attributs en vue de les segmenter par la suite.

2.2 Analyse globale de l'image

Si l'on se réfère à la littérature, la détection d'une région texturée ou uniforme est souvent réalisée en ne considérant que son écart-type [WC96]. Dans ce cas, une région d'une image est dite texturée si son écart-type est supérieur à un seuil. Or, l'écart-type ne tient pas compte de la distribution des niveaux de gris mais uniquement de la variation par rapport au niveau de gris moyen d'une région. La notion de texture étant liée à la résolution de l'observation de celle-ci, deux images ayant le même écart-type peuvent être l'une texturée et l'autre uniforme (composée de régions uniformes suffisamment grandes par rapport à la taille de l'image) comme le montre la figure FIG. 2.1.



(a) image non texturée

(b) image texturée

FIG. 2.1 – *Exemple de deux images ayant le même écart-type, l'une texturée (texture damier)et l'autre composée de 2 régions uniformes de taille importante*

Pour déterminer le caractère texturé ou uniforme d'une image, nous avons donc choisi une autre méthode basée sur la matrice de cooccurrence. Cette matrice est riche en informations et adaptée à la description de textures. La matrice de cooccurrence est une carte des transitions des niveaux de gris d'une image. Ainsi, si l'on considère par exemple la diagonale de cette matrice suivant 4 orientations et pour une distance de déplacement unité, on observe l'importance des transitions entre mêmes niveaux de gris dans toutes les directions. Une diagonale prépondérante par rapport aux autres éléments de la matrice traduit une homogénéité des transitions qui est caractéristique des zones uniformes. A contrario, elle est révélatrice de transitions désordonnées et donc de la présence de texture.

2.2.1 Caractère globale de l'uniformité d'une image

La matrice de cooccurrence, comme nous venons de le rappeler, est une carte des transitions des niveaux de gris d'une image. La diagonale de cette

matrice permet de quantifier l'importance des transitions entre mêmes niveaux de gris. Nous allons exploiter les valeurs de cette diagonale afin de mettre en évidence la présence des images dont la plupart des régions sont uniformes ou texturées.

On définit un attribut d'uniformité U correspondant à la moyenne des éléments de la diagonale de la matrice de cooccurrence pour une distance de déplacement de 1 et pour 4 orientations :

$$U = \frac{1}{nb} \sum_{j=1}^{nb} \sum_{i=1}^{NG} P_{1,\theta_j}(g_i,g_i) \tag{2.1}$$

où NG est le nombre de niveaux de gris de l'image, g_i un des niveaux de gris de l'image et $nb = 4$ le nombre d'orientations θ_j selon lesquelles on calcule la matrice de cooccurrence ($\theta_j = \frac{j\pi}{4}$,$j = 0..3$).

Le paramètre U permet de caractériser la fréquence de transitions entre mêmes niveaux de gris d'un pixel avec son voisinage 8-connexe noté V (cf. figure FIG. 2.2).

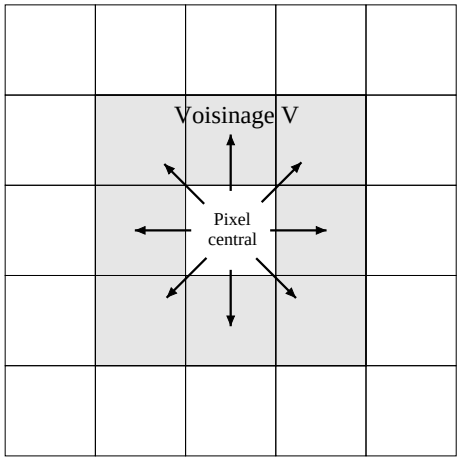


FIG. 2.2 – *Transition entre mêmes niveaux de gris dans un voisinage 8-connexe*

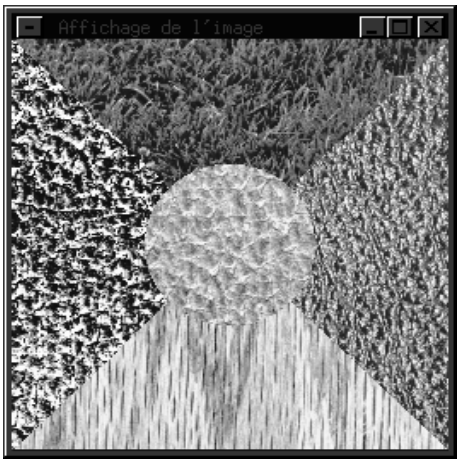
La matrice de cooccurrence est coûteuse en temps de calcul pour une image à plusieurs niveaux de gris. Afin de diminuer le nombre de niveaux de gris tout en conservant l'information significative, l'image est multi-seuillée. La méthode de multi-seuillage utilisée [KCC95] permet de sélectionner les niveaux de gris significatifs après une analyse locale de l'image. L'approche consiste à classer les points de l'image en comparant leurs niveaux de gris aux seuils déterminés par l'analyse d'un histogramme global. Celui-ci est construit à partir de pics significatifs issus d'histogrammes locaux. Cette procédure élimine l'information non pertinente afin d'homogénéiser les régions et de mettre en évidence les éléments les plus

caractéristiques des textures.

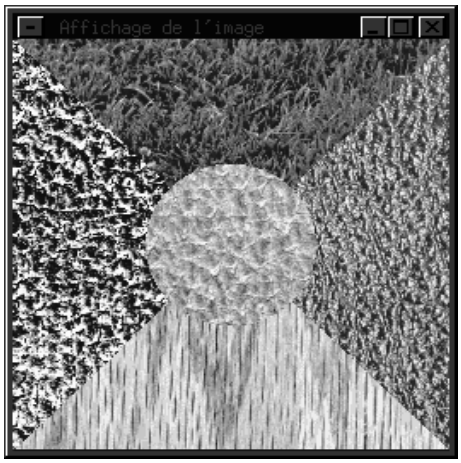
Pour montrer l'efficacité de cette méthode, nous présentons dans la figure FIG. 2.3 deux exemples d'images multi-seuillées ainsi que les images originales correspondantes.

(a) image originale (256 niveaux de gris)

(b) image multi-seuillée (23 niveaux de gris)



(c) image originale (256 niveaux de gris)



(d) image multi-seuillée (29 niveaux de gris)

FIG. 2.3 – *Résultats du Multiseuillage des images*

La matrice de cooccurrence résultante de l'image multi-seuillée est de dimension inférieure. Elle est donc plus exploitable d'une part parce qu'elle est plus robuste

vis à vis du bruit et, d'autre part, parce que seules les transitions significatives sont conservées.

Le paramètre $U(I)$ calculé sur l'image multi-seuillée globale I donne une indication sur le caractère global de l'uniformité d'une image. Néanmoins, ce paramètre dépend du nombre de niveaux de gris de l'image multi-seuillée. En effet, plus ce nombre est grand, moins la diagonale de la matrice de cooccurrence associée est prépondérante. Ainsi, deux résultats différents de multiseuillage de la même image peuvent donner deux valeurs distinctes du paramètre uniforme. Ceci met en évidence la difficulté de choisir un seuil pour décider si une image est globalement uniforme ou texturée. Dans la mesure où nous concevons un système de segmentation automatique, nous ne pouvons pas utiliser de seuil heuristique.

Afin de régler ce problème, nous estimons le seuil d'affectation pour une image multi-seuillée I en fonction du nombre de niveaux de gris de celle-ci.

Nous rappellons tout d'abord quelques notations :
- on note s un pixel de l'image I,
- on définit V_s le voisinage 8-connexe du pixel s et V_s^j son $j^{\text{ème}}$ voisin,
- on rappelle que I_s est le niveau de gris du pixel s,
- soit $\mathcal{V} = \{(I_{V_s^1},..,I_{V_s^8}) \ / \ \exists V_s^j \in V, I_{V_s^j} = I_s, \forall s \in I\}$ l'ensemble des configurations du voisinage du pixel s où au moins un de ses voisins a le même niveau de gris que celui-ci.

Le paramètre $U(I)$ correspond à la mesure de probabilité de l'évènement $\mathcal{V}$ pour l'image I. L'évènement $\overline{\mathcal{V}}$ correspond à l'ensemble des configurations où tous les pixels du voisinage du pixel s ont un niveau de gris différent de celui-ci. Sous l'hypothèse d'indépendance du niveau de gris d'un pixel avec celui de son voisin, la probabilité de l'évènement $\overline{\mathcal{V}}$ est égale à :

$$P[\overline{\mathcal{V}}] = (\frac{NG-1}{NG})^8$$

La valeur du paramètre d'uniformité estimée s'écrit donc :

$$\tilde{U}(I) = 1 - P[\overline{\mathcal{V}}] = 1 - (\frac{NG-1}{NG})^8$$

La valeur du paramètre $\tilde{U}(I)$ traduit la probabilité de transition de niveaux de gris d'un pixel avec un voisinage 8-connexe sous l'hypothèse d'indépendance des niveaux de gris de pixels. Lorsque cette hypothèse n'est pas satisfaite (cas d'une image composée de régions uniformes), la valeur $\tilde{U}(I)$ est sous-évaluée, d'où le critère de décision suivant :

$$\begin{cases} si\ U(I) < \tilde{U}(I) & \text{alors I est composée majoritairement de régions texturées} \\ sinon & \text{alors I est composée majoritairement de régions uniformes} \end{cases}$$

Pour vérifier la validité de ce critère, nous avons sélectionné un ensemble d'images réelles (télédétection, scènes d'extérieur) et synthétiques composées de régions diverses (voir figure FIG. 2.4). Certaines de ces images contiennent des zones fortement texturées.

Nous récapitulons dans le tableau TAB. 2.1 les valeurs de U et $\tilde{U}$ pour chaque image test. Les résultats montrent l'indépendance de l'affectation des images dans l'une des deux classes que nous nous sommes fixées (uniforme ou texturée) pour différents nombres de niveaux de gris de l'image multi-seuillée. L'analyse visuelle globale de l'image est conforme aux résultats obtenus automatiquement car toutes les images étiquetées comme majoritairement composées de régions texturées ou uniformes le sont effectivement.

Image	U	$\tilde{U}$	NG	type de région majoritaire
EXT1	43.95%	31%	22	uniforme
EXT1	55.2%	44%	14	uniforme
EXT2	61.17%	29%	23	uniforme
EXT2	64.8%	35%	19	uniforme
SYNT1	15.63%	24%	29	texturée
SYNT1	22.2%	36%	18	texturée
SYNT2	26.53%	40%	16	texturée
SYNT2	40.8%	61%	9	texturée
TELE1	40.71%	47%	13	texturée
TELE1	45.3%	56%	10	texturée
TELE2	32.95%	27%	25	uniforme
TELE2	52.9%	44%	14	uniforme

TAB. 2.1 – *Valeurs de U et $\tilde{U}$ pour chaque image test et pour différents nombres de niveaux de gris de l'image multi-seuillée*

L'information sur le caractère global de l'uniformité d'une image sera exploitée dans la phase suivante afin d'adapter au mieux les traitements postérieurs en fonction du contexte.

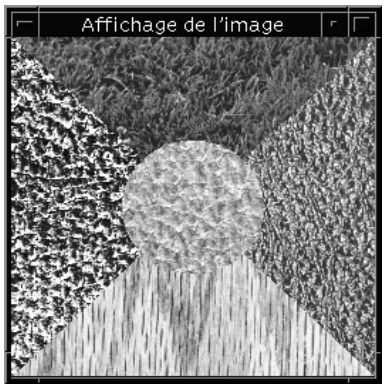
2.2.2 Localisation des zones texturées et uniformes

Cette phase de traitement a pour but de localiser les zones texturées et uniformes de l'image à segmenter. Cette partition des zones de l'image est importante dans la mesure où les méthodes de segmentation adaptées à chacune de ces zones sont relativement distinctes.

La notion de texture étant liée à la résolution d'observation, nous avons opté pour une approche multi-résolution de détection des zones uniformes et texturées

(a) scène d'extérieur : EXT1

(b) scène d'extérieur : EXT2



(c) image synthétique : SYNT1

(d) image synthétique : SYNT2

(e) image télédétection : TELE1

(f) image télédétection : TELE2

FIG. 2.4 – *Images utilisées pour valider le critère d'uniformité (taille* 256×256*)*

de l'image. La méthode consiste à calculer en chaque pixel de l'image le paramètre U défini précédemment à différentes résolutions en fonction du type d'image. Lorsque l'image est globalement uniforme, le paramètre uniforme est calculé sur plusieurs fenêtres W_i, $i = 1..4$ de taille $NW_i \times NW_i$ pixels ($NW_i = 2^{i-1} + 1$) de l'image I ayant $NCOL$ colonnes et $NLIG$ lignes (cf. FIG. 2.5). Dans le cas d'une image majoritairement composée de régions texturées, on calcule ce paramètre en considérant 3 tailles de fenêtres ($NW_i = 2^{i+2} + 1$, $i = 1{,}3$) plus grandes pour prendre en compte l'aspect texturé de l'image.

Soit s un pixel de l'image I, le paramètre $U(W_i)$ est calculé à partir de la matrice de cooccurrence sur la fenêtre W_i centrée en s. Dans le cas où le pixel $s \in Bord = I - \check{I}$ ($\check{I} = [\frac{NW_i-1}{2}, NLIG - \frac{NW_i-1}{2}] \times [\frac{NW_i-1}{2}, NCOL - \frac{NW_i-1}{2}]$), le paramètre U du site est pris égal à celui du pixel $s' \in \check{I}$ le plus proche.

2.2.2.1 Critère de partition des zones de l'image

Un pixel de l'image appartient à une zone uniforme si la moyenne des paramètres $U(W_i)$ associée est supérieure à la valeur $\tilde{U}(I)$ définie précédemment sinon il appartient à une région texturée.

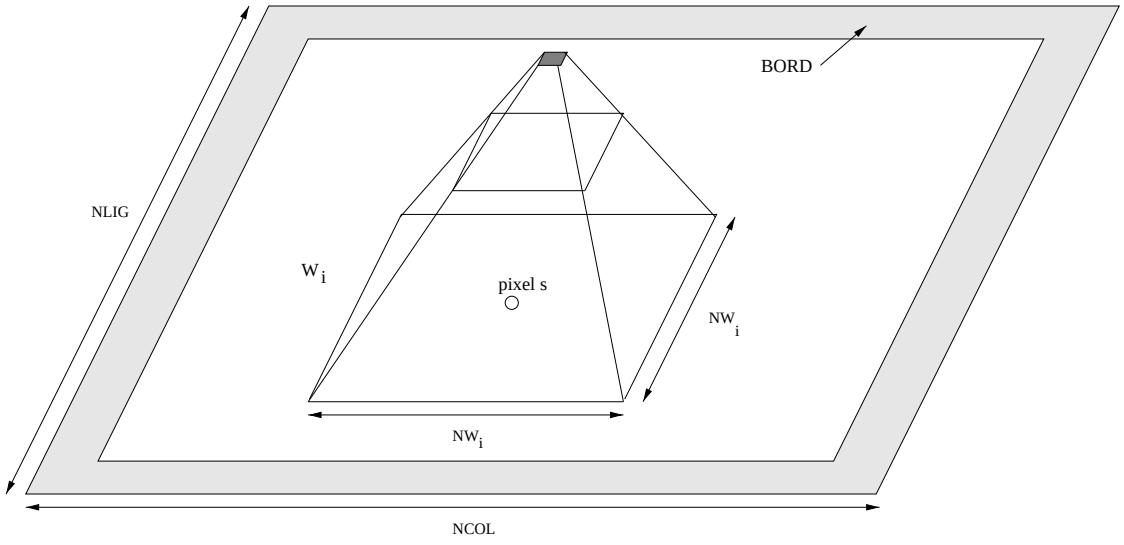


FIG. 2.5 – *Détection des zones texturées et uniformes de l'image à segmenter*

Néanmoins, il peut se produire des erreurs d'affectation de différents types :

- des pixels appartenant à une zone uniforme mais proches d'une frontière d'une région texturée peuvent être étiquetés comme texturés.

– la présence d'une région uniforme contenant peu de pixels à l'intérieur d'une région texturée n'est pas justifiée. Il faut donc ré-affecter les pixels de cette zone dans la classe texturée.

L'erreur d'affectation d'un pixel appartenant a priori à une zone uniforme est commise lorsqu'il est situé à moins de $\frac{NW_i-1}{2}$ pixels d'une zone texturée. Dans ce cas, le paramètre uniforme sera faible. Il faut noter que le problème ne se pose pas lorsqu'un pixel se trouve proche d'une transition entre deux zones uniformes puisque le paramètre uniforme sera élevé.

On note I_{MS} l'image multi-seuillée de l'image I et I_D l'image résultat de la localisation des zones uniformes et texturées de l'image I. L'image I_D est définie de la façon suivante :

$$\forall s \in I, \;\; I_D(s) = \begin{cases} 255 & \text{le pixel s est dans une région uniforme} \\ 0 & sinon \end{cases}$$

L'image I_D est analysée afin de ré-affecter toute région contenant un nombre de pixels inférieur à 100 pixels. Ceci permet de ne conserver pour l'étape de segmentation que des zones de l'image de tailles significatives.

Afin de régler le problème de la mauvaise affectation des pixels appartenant à une zone frontière entre régions uniforme et texturée, nous appliquons l'opérateur morphologique "fermeture". Cette opération consiste à appliquer successivement une dilatation des zones uniformes par un élément structurant (un disque de rayon 3 pixels) puis une érosion par le même élément. Après ce traitement les zones frontières entre régions uniformes alors étiquetées comme texturées seront labellisées uniformes.

Une fois le processus de fermeture des régions de l'image I_D achevé, l'image à segmenter I est partitionnée en deux classes l'une contenant les zones uniformes de l'image et l'autre celles texturées.

2.2.2.2 Résultats expérimentaux

Nous présentons tout d'abord, pour chaque image test, les valeurs moyennes du paramètre uniforme U multi-résolution pour chaque pixel suivant plusieurs résolutions (cf. figures FIG. 2.6(b), FIG. 2.7(b), FIG. 2.8(b), FIG. 2.9(b), FIG. 2.10(b) et FIG. 2.11(b)) ainsi que l'histogramme associé. Le paramètre uniforme est présenté sous forme d'une image (codée sur 256 niveaux) où le niveau de gris 255 correspond à la valeur maximale du paramètre uniforme. L'histogramme de cette image permet de mettre en évidence le caractère textural global et local de l'image. En effet, la présence de pics pour des fortes (resp faibles) valeurs du paramètre est révélatrice de la présence de zones uniformes (resp texturées) dans l'image.

En ce qui concerne les résultats finaux (cf. figures FIG. 2.6(d), FIG. 2.7(d), FIG. 2.8(d), FIG. 2.9(d), FIG. 2.10(d) et FIG. 2.11(d)), on note une bonne détection des zones uniformes et texturées. L'image TELE1 est principalement composée de régions texturées alors que l'image TELE2 ne contient pratiquement que des régions uniformes. Dans ces deux cas, la détection de ces zones est visuellement correcte.

Les images synthétiques ne sont composées que de zones texturées comme le montre le résultat final du traitement pour chaque image. Certaines zones détectées comme uniformes correspondent à des régions faiblement texturées de l'image. Par ailleurs, on remarque qu'une partie de la région de l'image SYNT2 située à droite a été détectée comme région faiblement texturée. Ceci s'explique par une faible variation des niveaux de gris, se traduisant par une valeur élevée du paramètre uniforme.

Les images de scène d'extérieur sont les plus intéressantes car elles sont composées à la fois de régions texturées et uniformes. Dans les deux cas (image EXT1 et EXT2) les zones uniformes ou faiblement texturées sont correctement détectées. Dans le cas de l'image EXT1, l'arbre, les bordures et les fenêtres de la maison sont détectés comme texturés. Les autres zones de l'image sont détectées comme uniformes (toit et pelouse faiblement texturés..). Une partie de la pelouse de l'image EXT2 a été détectée comme texturée (partie contenant des variations plus discernables).

2.3 Analyse locale de l'image

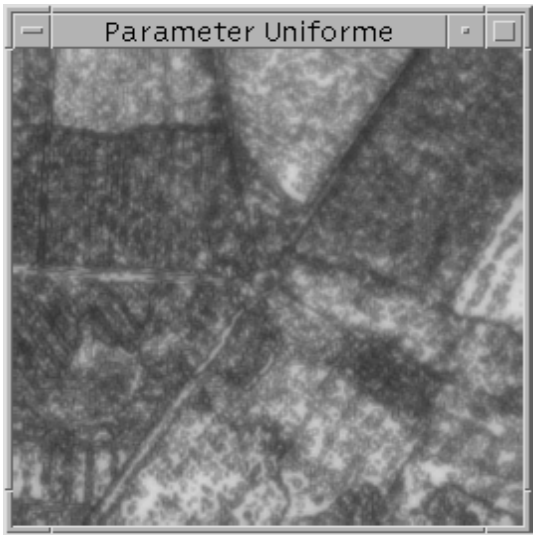
Afin de faciliter le choix de la méthode de segmentation adaptée à chaque type de zone (uniforme, texturée), nous essayons d'affiner cette description des régions de l'image.

En ce qui concerne les régions uniformes, il n'est pas nécessaire d'avoir d'informations supplémentaires dans la mesure où la segmentation de ce type de région ne pose pas de problème particulier. Il serait possible néanmoins d'extraire sur ces régions de multiples informations tels que le contraste, la surface, etc..

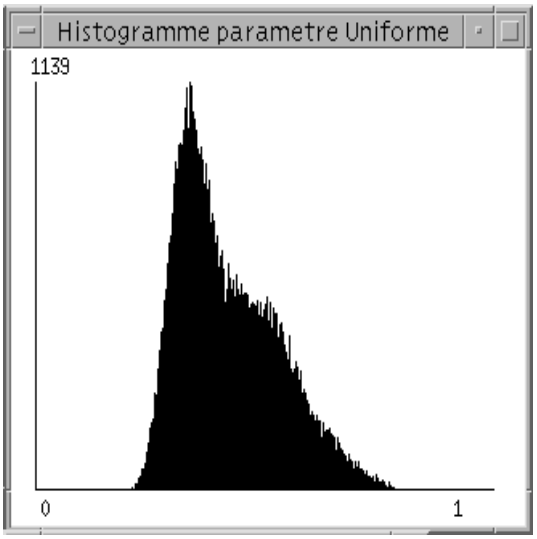
Par contre dans le cas de régions texturées, il existe beaucoup de méthodes de segmentation où chacune d'entre elles est adaptée à un ensemble de textures plus ou moins restreint. Afin de réaliser une bonne adéquation entre méthodes de segmentation et régions texturées de l'image extraites lors de l'étape précédente, il est nécessaire d'extraire de multiples informations sur la ou les textures présentes à l'intérieur d'une image.

Nous proposons dans un premier temps d'analyser un certain nombre d'attributs existant dans la littérature pour quantifier leur redondance et leur pouvoir discri-

(a) image télédétection : TELE1



(b) résultat intermédiaire (paramètre uniforme)

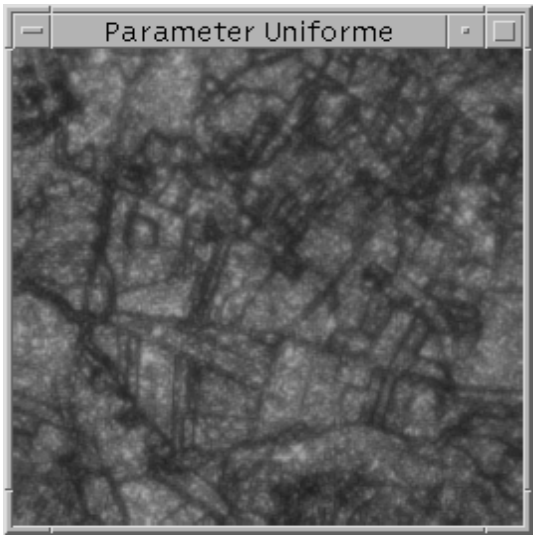


(c) histogramme paramètre Uniforme

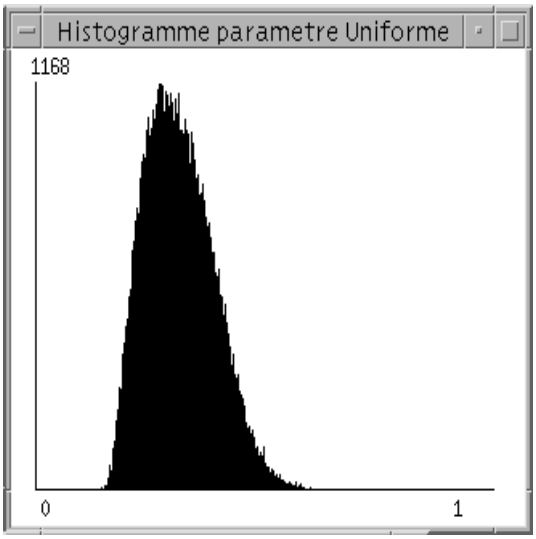
(d) Résultat final

FIG. 2.6 – *Détection des zones uniformes de l'image TELE1 : les zones noires correspondent aux régions texturées*

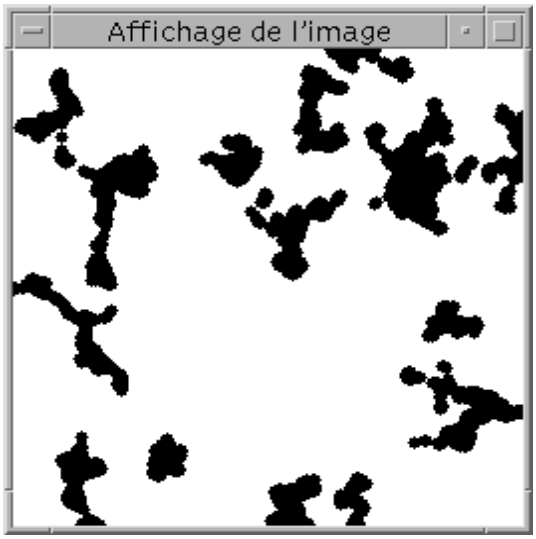
(a) image télédétection : TELE2



(b) résultat intermédiaire (paramètre uniforme)

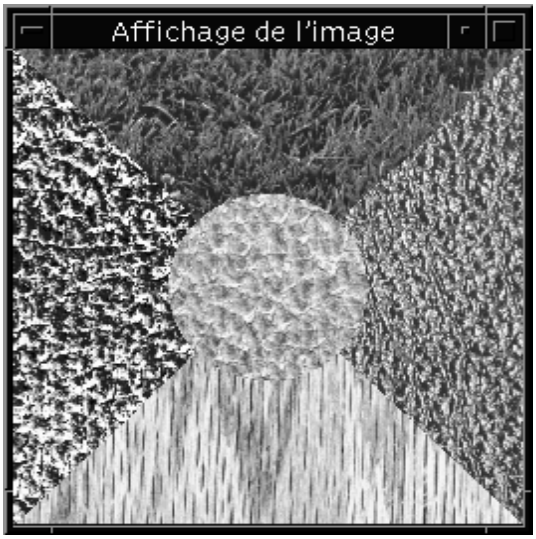


(c) histogramme paramètre Uniforme

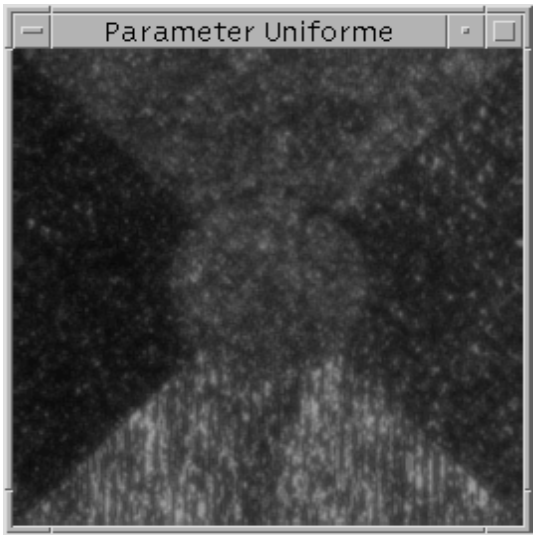


(d) Résultat final

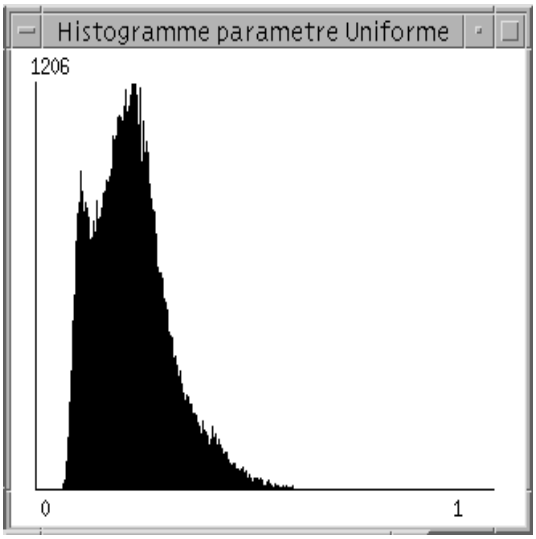
FIG. 2.7 – *Détection des zones uniformes de l'image TELE2 : les zones noires correspondent aux régions texturées*



(a) image synthétique : SYNT1



(b) résultat intermédiaire (paramètre uniforme)

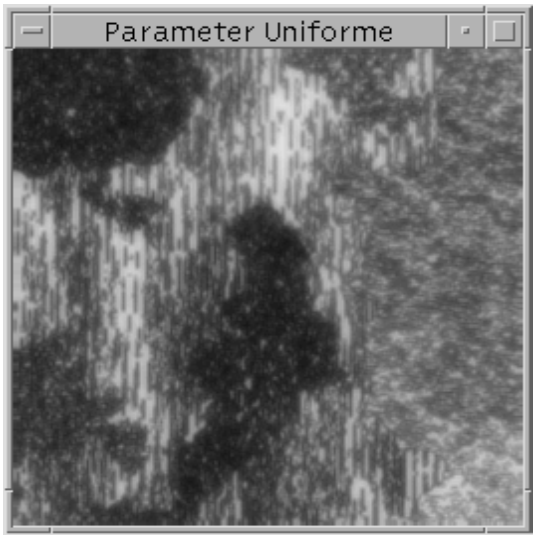


(c) histogramme paramètre Uniforme

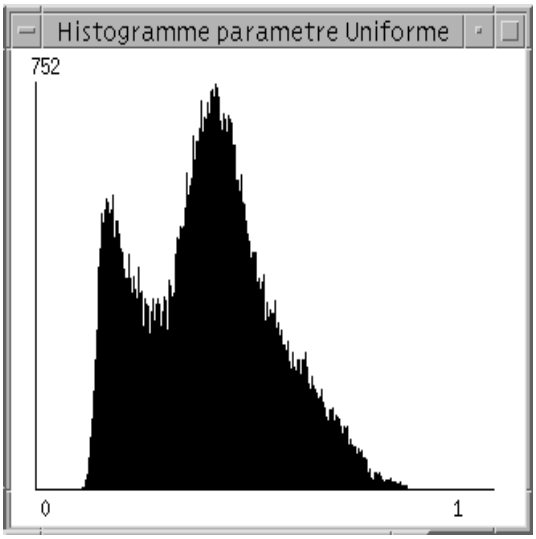
(d) Résultat final

FIG. 2.8 – *Détection des zones uniformes de l'image SYNT1 : les zones noires correspondent aux régions texturées*

(a) image synthétique : SYNT2



(b) résultat intermédiaire (paramètre uniforme)



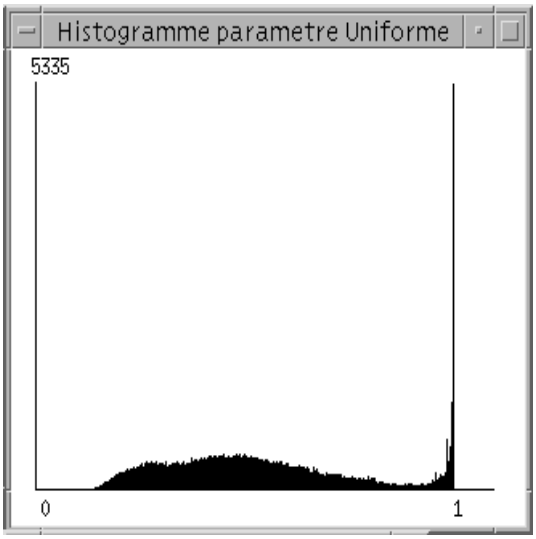
(c) histogramme paramètre Uniforme

(d) Résultat final

FIG. 2.9 – *Détection des zones uniformes de l'image SYNT2 : les zones noires correspondent aux régions texturées*

(a) image scène d'extérieur : EXT1

(b) résultat intermédiaire (paramètre uniforme)



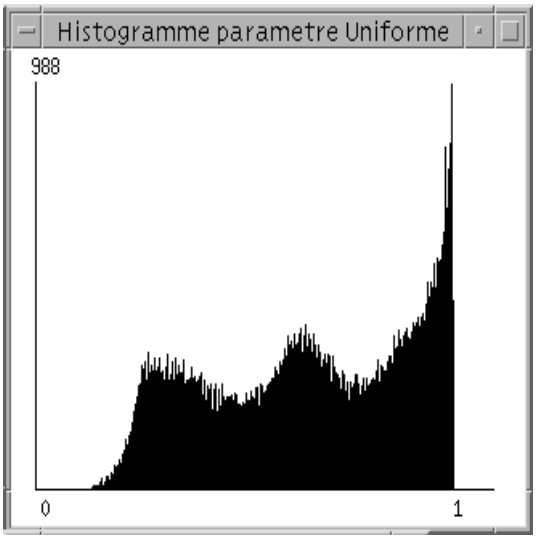
(c) histogramme paramètre Uniforme

(d) Résultat final

FIG. 2.10 – *Détection des zones uniformes de l'image EXT1 : les zones noires correspondent aux régions texturées*

(a) image scène d'extérieur : EXT2

(b) résultat intermédiaire (paramètre uniforme)



(c) histogramme paramètre Uniforme

(d) Résultat final

FIG. 2.11 – *Détection des zones uniformes de l'image EXT2 : les zones noires correspondent aux régions texturées*

minant. Ensuite, nous proposons de nouveaux paramètres de texture permettant de répertorier les textures en fonction de leur caractère déterministe ou aléatoire, et de leur granularité.

2.3.1 Analyse d'attributs de texture classiques

Afin d'obtenir des informations sur la texture, nous avons retenu les descripteurs de texture les plus performants [RC97] et ne nécessitant aucune hypothèse d'utilisation :

- attributs de cooccurrence [HSD73],
- attributs de longueur de plages [Lia94],
- attributs des histogrammes locaux [RBR85],
- attributs des extrema locaux [HCR92],
- attributs de l'intégrale curviligne [HCR92].

Dans la mesure où ces descripteurs sont couramment utilisés en classification de textures, ils doivent permettre la distinction de deux textures différentes en considérant plusieurs propriétés texturales.

Vu l'importance du nombre de paramètres qui peuvent être utilisés (cf. Annexes A)(30 attributs : 15 de la matrice de cooccurrence, 5 de la matrice de longueur de plages, 2 des histogrammes locaux, 4 des extrema locaux et 4 de l'intégrale curviligine), nous allons dans une étape préliminaire étudier la redondance ou la corrélation des attributs de texture sélectionnés de même type puis de types différents. Ceci permet donc de réduire l'espace de mesures car l'utilisation conjointe d'attributs redondants n'apporte aucune information complémentaire sur la texture.

2.3.1.1 Analyse de la corrélation des attributs

Cette analyse a pour objectif d'évaluer la redondance d'information entre attributs afin de ne conserver que les plus significatifs [RC99]. Afin d'estimer cette redondance, nous avons analysé la matrice des facteurs de corrélation de chaque type d'attribut.

Pour déterminer la robustesse de la méthode, nous avons classé différentes textures en utilisant dans un premier temps l'ensemble des paramètres puis les attributs après réduction.

Un attribut a est caractérisé par un vecteur $\mathbf{a}$ de dimension n tel que $\mathbf{a}_k$ $1 < k < n$ est la valeur de cet attribut pour l'échantillon k. La corrélation entre les différents attributs est obtenue à partir de la matrice des coefficients de corrélation dont chaque élément est :

$$Correl(a^i,a^j) = \frac{Cov(\mathbf{a}^i,\mathbf{a}^j)}{\sqrt{Cov(\mathbf{a}^i,\mathbf{a}^i)Cov(\mathbf{a}^j,\mathbf{a}^j)}}$$

où $Cov(\mathbf{a}^i,\mathbf{a}^j)$ est la covariance de l'attribut a^i et a^j.

Cette matrice est symétrique et à éléments dans l'intervalle $[-1,1]$. Le facteur de corrélation $Correl(a^i,a^j)$ traduit la redondance entre les deux attributs a^i et a^j; celle-ci est d'autant plus forte que la valeur $|Correl(a^i,a^j)|$ est proche de 1. Par la suite, nous allons considérer uniquement la valeur absolue du facteur de corrélation pour quantifier la redondance de deux attributs. En effet, le but de l'étude est ici d'identifier les attributs n'apportant aucune information supplémentaire pour la classification de textures. Or, réaliser une classification à partir de deux attributs $(\mathbf{a}^1,\mathbf{a}^2)$ ou $(\mathbf{a}^1,\text{-}\mathbf{a}^2)$ conduira au même résultat. Par conséquent, deux attributs sont complémentaires ou pertinents si la valeur absolue de leur facteur de corrélation est non négligeable.

Afin d'évaluer la corrélation entre les différents attributs de texture présentés précédemment, nous avons utilisé une mosaïque contenant 50 échantillons de 21 textures différentes représentatives de l'album de Brodatz [Bro56] de taille 64×64 pixels (cf. FIG. 2.12). Certaines textures appartenant à la même classe ont des contrastes différents. A partir de cette mosaïque, nous avons extrait des imagettes de taille 16×16 pixels ce qui correspond à 800 échantillons pour chaque attribut de texture. Avant de donner les différents résultats, nous précisons les différents attributs utilisés (cf. Annexes A) :

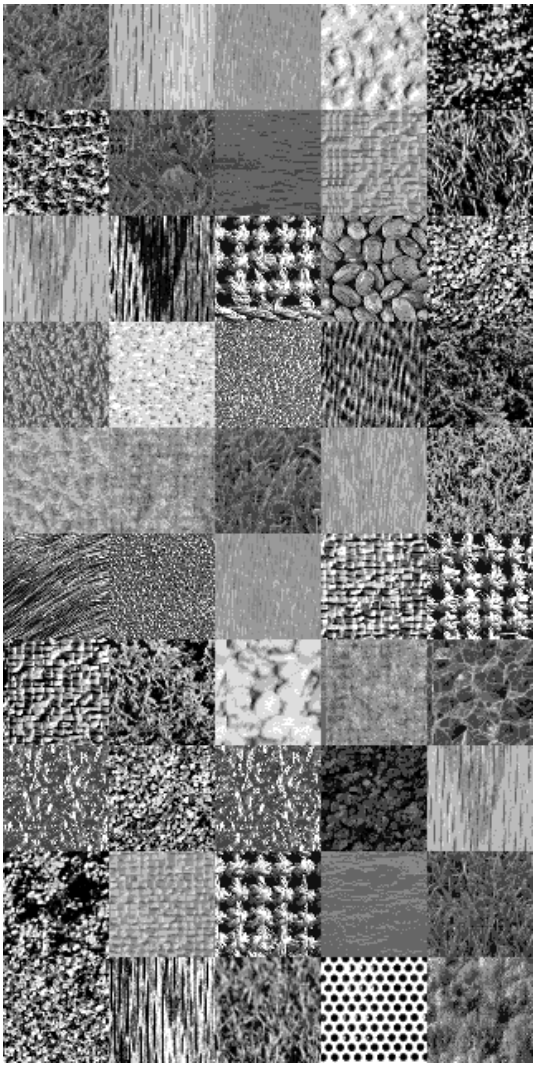
- **attributs de la matrice de cooccurrence** :
  - second moment angulaire noté *cooc*1.
  - contraste noté *cooc*2.
  - corrélation noté *cooc*3.
  - variance noté *cooc*4.
  - moment des différences inverses noté *cooc*5.
  - moyennes des sommes noté *cooc*6.
  - variances des sommes noté *cooc*7.
  - entropie des sommes noté *cooc*8.
  - entropie noté *cooc*9.
  - variance des différences noté *cooc*10.
  - entropie de la différence noté *cooc*11.
  - première information sur la mesure de corrélation noté *cooc*12.
  - deuxième information sur la mesure de corrélation noté *cooc*13.
  - uniforme noté *cooc*14.
  - information contour noté *cooc*15.
- **attributs de la matrice de longueur de plages** :
  - importance des courtes plages noté *ldp*1.
  - importance des longues plages noté *ldp*2.

  - distribution des niveaux de gris noté $ldp3$.
  - distribution des plages noté $ldp4$.
  - pourcentage de plages noté $ldp5$.
- **attributs des histogrammes locaux** :
  - module moyen noté $hist1$.
  - phase moyenne noté $hist2$
- **attributs des extrema locaux** : 4 attributs correspondant au nombre d'extrema moyens dans chacune des directions ($k.\pi/4$, $k = 0{,}3$) (ext1, ext2, ext3, ext4) sont extraits.
- **attributs de l'intégrale curviligne** : 4 attributs contenant l'intégrale curviligne moyenne dans 4 directions ($\theta = k.\frac{\pi}{4}$, $k = 0{,}3$) pour une distance de balayage donnée d (curv1, curv2, curv3, curv4) sont calculés.

Nous donnons la matrice de corrélation totale des différents attributs (cf. TAB. 2.2). Ce tableau de corrélation donne d'une part, la corrélation entre attributs de même type tels que ceux issus de la matrice de cooccurrence et d'autre part, la corrélation entre attributs de différents types.

Nous avons mis en évidence les facteurs de corrélation supérieur à 0.95 dans le le tableau TAB. 2.2 pour détecter des attributs fortement corrélés.

- **Remarques sur les attributs de même type**:
- **attributs de la matrice de cooccurrence** : 9 facteurs de corrélation sur 105 sont strictement supérieurs à 0.95 :
  - le paramètre cooc4 est très fortement corrélé à l'attribut cooc7. Le premier quantifie la variance des transitions des niveaux de gris de l'image et le second la variance des contrastes des niveaux de gris,
  - les attributs cooc8 (entropie matrice de cooccurrence) et cooc9 (entropie de sommes) sont corrélés, ce qui s'explique assez facilement. En effet, le paramètre cooc8 mesure la régularité des transitions des niveaux de gris de la matrice de cooccurrence et l'attribut cooc9 celle des contrastes des niveaux de gris,
  - l'attribut cooc10 (variance des différences) et cooc1 (second moment angulaire) sont redondants. Le paramètre cooc1 mesure l'homogénéité des transitions des niveaux dans l'image et cooc10 la variance du contraste des niveaux de gris de l'image,
  - l'attribut cooc11 est fortement corrélé avec 3 attributs (cooc5, cooc9, cooc10). L'attribut cooc11 mesure l'homogénéité des transitions de niveaux de l'image. Or, le paramètre cooc5 quantifie l'importance des transitions entre niveaux de gris proches, ce qui quantifie la régularité des transitions entre tous les niveaux de gris. L'explication pour les attributs cooc9 et cooc10 est identique,



D9	D68	D37	D57	D29
D24	D9	D70	D84	D9
D68	D68	D77	D74	D29
D24	D32	D4	D53	D29
D38	D19	D9	D37	D9
D93	D4	D37	D84	D77
D84	D9	D23	D19	D112
D24	D29	D24	D28	D68
D29	D84	D77	D70	D9
D29	D68	D9	D66	D2

FIG. 2.12 – *Mosaïque de textures*

	cooc1	**cooc2**	**cooc3**	**cooc4**	**cooc5**	**cooc6**	**cooc7**	**cooc8**	**cooc9**	**cooc10**	**cooc11**	**cooc12**	**cooc13**	**cooc14**	**cooc15**
cooc1		0.28	0.31	0.24	0.88	0.32	0.26	0.95	0.90	**0.96**	0.91	0.47	0.36	0.89	0.95
cooc2			0.29	0.93	0.43	0.37	0.92	0.29	0.21	0.32	0.35	0.22	0.17	0.30	0.24
cooc3				0.04	0.36	0.59	0.06	0.23	0.23	0.36	0.32	0.48	0.18	0.39	0.36
cooc4					0.35	0.61	**0.99**	0.27	0.19	0.25	0.29	0.06	0.11	0.21	0.16
cooc5						0.23	0.36	0.89	0.93	0.95	**0.96**	0.72	0.51	**0.99**	0.92
cooc6							0.58	0.26	0.29	0.32	0.25	0.39	0.21	0.36	0.42
cooc7								0.30	0.21	0.27	0.31	0.06	0.12	0.23	0.19
cooc8									**0.96**	0.94	0.95	0.44	0.42	0.89	0.95
cooc9										0.94	**0.97**	0.56	0.49	0.95	0.95
cooc10											**0.97**	0.63	0.44	**0.96**	0.95
cooc11												0.63	0.49	**0.96**	0.93
cooc12													0.38	0.74	0.57
cooc13														0.50	0.45
cooc14															0.94
ldp1	0.85	0.27	0.38	0.19	**0.97**	0.33	0.20	0.87	0.94	0.94	**0.96**	0.73	0.52	**0.98**	0.90
ldp2	0.95	0.26	0.28	0.23	0.79	0.24	0.25	0.87	0.81	0.90	0.83	0.45	0.23	0.82	0.85
ldp3	0.35	0.06	0.11	0.02	0.51	0.03	0.03	0.57	0.69	0.45	0.60	0.20	0.42	0.51	0.48
ldp4	0.78	0.28	0.38	0.17	**0.96**	0.33	0.19	0.80	0.91	0.89	0.92	0.77	0.55	**0.97**	0.88
ldp5	0.89	0.30	0.39	0.21	**0.99**	0.36	0.22	0.89	0.95	**0.96**	**0.96**	0.74	0.50	**1.00**	0.94
hist1	0.03	0.12	0.13	0.07	0.04	0.08	0.07	0.03	0.00	0.04	0.05	0.08	0.03	0.04	0.03
hist2	0.03	0.11	0.07	0.08	0.03	0.06	0.08	0.01	0.02	0.03	0.02	0.08	0.02	0.02	0.03
ext1	0.31	0.20	0.20	0.15	0.51	0.03	0.14	0.30	0.41	0.40	0.43	0.45	0.33	0.49	0.36
ext2	0.43	0.26	0.19	0.21	0.62	0.05	0.21	0.44	0.53	0.53	0.55	0.52	0.37	0.60	0.48
ext3	0.50	0.18	0.11	0.17	0.66	0.10	0.17	0.55	0.66	0.60	0.65	0.57	0.36	0.66	0.57
ext4	0.44	0.22	0.22	0.17	0.59	0.08	0.17	0.43	0.51	0.52	0.53	0.49	0.37	0.58	0.47
curv1	0.29	0.32	0.45	0.14	0.31	0.13	0.14	0.39	0.47	0.33	0.40	0.06	0.21	0.32	0.30
curv2	0.29	0.32	0.46	0.13	0.30	0.15	0.13	0.39	0.46	0.33	0.40	0.05	0.21	0.31	0.29
curv3	0.28	0.32	0.45	0.14	0.31	0.14	0.14	0.39	0.46	0.33	0.40	0.06	0.21	0.32	0.29
curv4	0.28	0.32	0.47	0.13	0.31	0.15	0.13	0.39	0.46	0.33	0.39	0.05	0.21	0.31	0.29

	ldp1	**ldp2**	**ldp3**	**ldp4**	**ldp5**	**hist1**	**hist2**	**ext1**	**ext2**	**ext3**	**ext4**	**curv1**	**curv2**	**curv3**	**curv4**
ldp1		0.78	0.58	**0.97**	**0.98**	0.03	0.01	0.50	0.60	0.67	0.58	0.38	0.37	0.37	0.37
ldp2			0.23	0.69	0.82	0.03	0.02	0.26	0.38	0.44	0.39	0.26	0.26	0.26	0.26
ldp3				0.58	0.51	0.08	0.11	0.28	0.35	0.54	0.31	0.63	0.63	0.63	0.63
ldp4					**0.97**	0.02	0.01	0.54	0.63	0.68	0.60	0.34	0.33	0.33	0.33
ldp5						0.04	0.02	0.49	0.60	0.66	0.58	0.32	0.31	0.32	0.31
hist1							0.84	0.05	0.00	0.00	0.03	0.12	0.11	0.11	0.11
hist2								0.03	0.02	0.01	0.05	0.10	0.10	0.10	0.11
ext1									0.82	0.37	0.60	0.13	0.13	0.14	0.14
ext2										0.45	0.58	0.21	0.19	0.20	0.21
ext3											0.43	0.41	0.40	0.41	0.40
ext4												0.23	0.21	0.22	0.21
curv1													**0.98**	**0.96**	0.94
curv2														**0.98**	0.94
curv3															**0.98**
curv4															

TAB. 2.2 – *Corrélation des différents paramètres de textures*

  - le paramètre cooc14 est redondant avec trois autres attributs (cooc5, cooc10, cooc11). Il mesure l'importance des transitions entre mêmes niveaux de gris. Cette information peut être obtenue par le paramètre cooc5 mesurant l'importance des transitions entre niveaux de gris proches. Comme nous avons montré que le paramètre cooc11 est fortement corrélé aux attributs cooc5 et cooc10, il convient que le paramètre cooc14 soit corrélé aux attributs cooc10 et cooc11.
- **attributs de longueur de plages** : 3 facteurs de corrélation sur 10 sont supérieurs à 0.95. Ils concernent les attributs ldp1, ldp4 et ldp5. Les paramètres ldp1 et ldp4 sont par définition redondants puisque l'un quantifie l'importance des plages courtes et l'autre le pourcentage de plages par unité de surface. Ainsi, un pourcentage élevé de plages par unité de surface traduit une prépondérance de plages courtes. De même, le paramètre ldp4 quantifie la non-uniformité des plages, ce qui est lié à l'importance des plages courtes et donc à ldp1. On obtient alors la justification de la redondance de ldp1 et ldp5.
- **attributs des histogrammes locaux** : le seul facteur de corrélation à considérer (2 attributs) est inférieur à 0.95.
- **attributs des extrema locaux** : aucun facteur de corrélation n'est supérieur à 0.95.
- **attributs de l'intégrale curviligne** : 4 des 6 facteurs de corrélation sont strictement supérieurs à 0.95. Ceci montre qu'il n'est pas nécessaire de calculer cet attribut suivant quatre orientations (la base d'échantillons est pourtant bien constituée de textures anisotropes).

• **Remarques sur les attributs de différents types**:

En ce qui concerne la corrélation entre attributs de types différents, on peut noter que seuls 9 facteurs de corrélation sur 107 possibles sont strictement supérieurs à 0.95. Ces facteurs de corrélation concernent uniquement les attributs de longueur de plages et ceux de la cooccurrence :

- l'attribut ldp1 est fortement corrélé à 3 attributs (cooc5, cooc11, cooc14). Le paramètre ldp1 quantifie l'importance des courtes plages, ce qui est fortement lié à l'importance des transitions entre niveaux de gris proches mesurée par l'attribut cooc5. La corrélation du paramétre ldp1 avec les attributs cooc11 et cooc14 s'explique compte tenu du fait que nous avons montré précedemment que ces paramètres sont fortement dépendants du paramètre cooc5.
- le paramètre ldp4 est fortement corrélé aux attributs cooc5 et cooc14. Il mesure en effet la non-uniformité de la répartition des plages, ce qui est fortement dépendant de l'importance des transitions entre niveaux de gris proches quantifiée par le paramètre cooc5.
- le paramètre ldp5 est fortement corrélé aux attributs cooc5, cooc10, cooc11 et cooc14. Ces facteurs de corrélation s'expliquent comme précédemment. Néanmoins, on constate que le facteur de corrélation entre cooc14 et ldp5 vaut 1

signifiant que ces attributs varient de la même façon. Or, si l'on regarde de plus près la formule de ces attributs, on remarque que ces deux attributs sont égaux si la longueur de la plage maximale est égale à la taille de fenêtre d'analyse. On peut en effet montrer dans ce cas que :

$$P_{1,\theta}(i,i) = \sum_{l=1}^{L_{max}} PL(i,l,\theta) \ ,\forall i = 1..NG$$

Cette étude sur la corrélation des différents attributs de texture montre que certains d'entre eux sont fortement corrélés. Nous avons calculé la corrélation moyenne de chaque type d'attribut (cf. TAB. 2.3) ainsi que la corrélation moyenne entre attributs de différents types.

attributs	COOC	LDP	EXT	HIST	CURV
COOC	0.51	0.58	0.05	0.38	0.28
LDP		0.71	0.04	0.50	0.38
EXT			0.54	0.02	0.24
HIST				0.84	0.10
CURV					0.96

TAB. 2.3 – *Corrélation moyenne de chaque type de descripteurs de texture*

On remarque que la plupart des attributs sont relativement complémentaires dans la mesure où la corrélation moyenne entre chaque type d'attribut est faible. Il faut noter néanmoins que les attributs de la matrice de longueur de plages n'apportent pas beaucoup d'informations supplémentaires par rapport à ceux de la matrice de cooccurrence ou des histogrammes locaux. Hormis la redondance entre certains attributs de la matrice de cooccurrence et ceux de la matrice de longueur de plages, tous les descripteurs retenus sont relativement complémentaires. Ceci montre l'intérêt de combiner ces différents attributs dans une phase d'analyse de textures puisqu'ils apportent des informations de nature différentes.

Nous allons dans un premier temps sélectionner les attributs de texture non redondants issus du même type, puis globalement.

Avant de réduire l'espace de mesure, nous avons cherché à quantifier l'apport d'information pour chaque type d'attributs. Cette analyse complémentaire à la précédente mettant en évidence la redondance globale des paramètres (cf. FIG. 2.13), s'appuie sur les valeurs propres de la matrice de corrélation.

Nous définissons λ_i, $\forall i = 1..m$ les valeurs propres de la matrice de corrélation $Correl$ pour m attributs. Les vecteurs propres v_i, $\forall i = 1..m$ associés satisfont la relation suivante :

$$Correl.v_i = \lambda_i.v_i, \ \forall i = 1..m$$

Dans la mesure où la matrice *Correl* est symétrique et à valeurs réelles, il est possible de trouver des vecteurs propres orthogonaux, c'est à dire vérifiant cette propriété :

$$< v_i, v_j >= 0, \forall i \neq j$$

où $< .,. >$ est le produit scalaire.

La matrice de corrélation peut donc s'écrire sous la forme :

$$Correl = V.D.V^T$$

avec

$$D = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{pmatrix} \quad \text{et} \quad V = \begin{pmatrix} v_1^0 & \cdots & v_1^0 \\ \vdots & & \\ v_m^0 & \cdots & v_m^0 \end{pmatrix}$$

La transformée de Karhunen-Loève discrète permet de décorréler un ensemble d'attributs a^i, $i = 1..m$ étant donnés la matrice V et le vecteur d'attribut moyen $C = \frac{1}{m}\sum_{i=1}^{m} \mathbf{a}^i$. Les attributs obtenus par cette transformation sont notés $\overset{\perp}{\mathbf{a}}^i$, $i = 1..m$ et sont calculés comme suit :

$$\overset{\perp}{\mathbf{a}}^i = V.(\mathbf{a}^i - C)$$

L'ensemble des attributs $\overset{\perp}{\mathbf{a}}^i$, $i = 1..m$ forme une base orthogonale de dimension m.

On observe sur la figure FIG. 2.13 que certains attributs ont une valeur propre négligeable par rapport aux autres. Cela signifie que ces paramètres jouent un rôle négligeable dans la description de l'individu (une texture dans notre cas). La transformée de Karhunen-Loève permet aussi de réduire l'espace des paramètres en ne conservant que les attributs ayant une valeur propre significative. Le principe de sélection des attributs consiste à ne conserver un attribut a^k que s'il induit une information non négligeable, c'est-à-dire, si $\lambda_k / \sum_i \lambda_i > \delta$. La valeur δ correspond au pourcentage minimal toléré de la pertinence d'un attribut par rapport aux autres. Ainsi, étant donnée une valeur de δ, il est possible de réduire l'espace d'attributs en conséquence. Cette méthode permet effectivement de réduire l'espace des attributs mais présente le désavantage de nécessiter le calcul de l'ensemble des attributs. Afin de pallier cet inconvénient, nous proposons une approche de sélection des attributs.

2.3.1.2 Réduction de l'espace des attributs

La matrice de corrélation permet de sélectionner un sous-ensemble d'attributs ayant une redondance mutuelle admissible. Un attribut $\mathbf{a}^i$ est dit redondant par

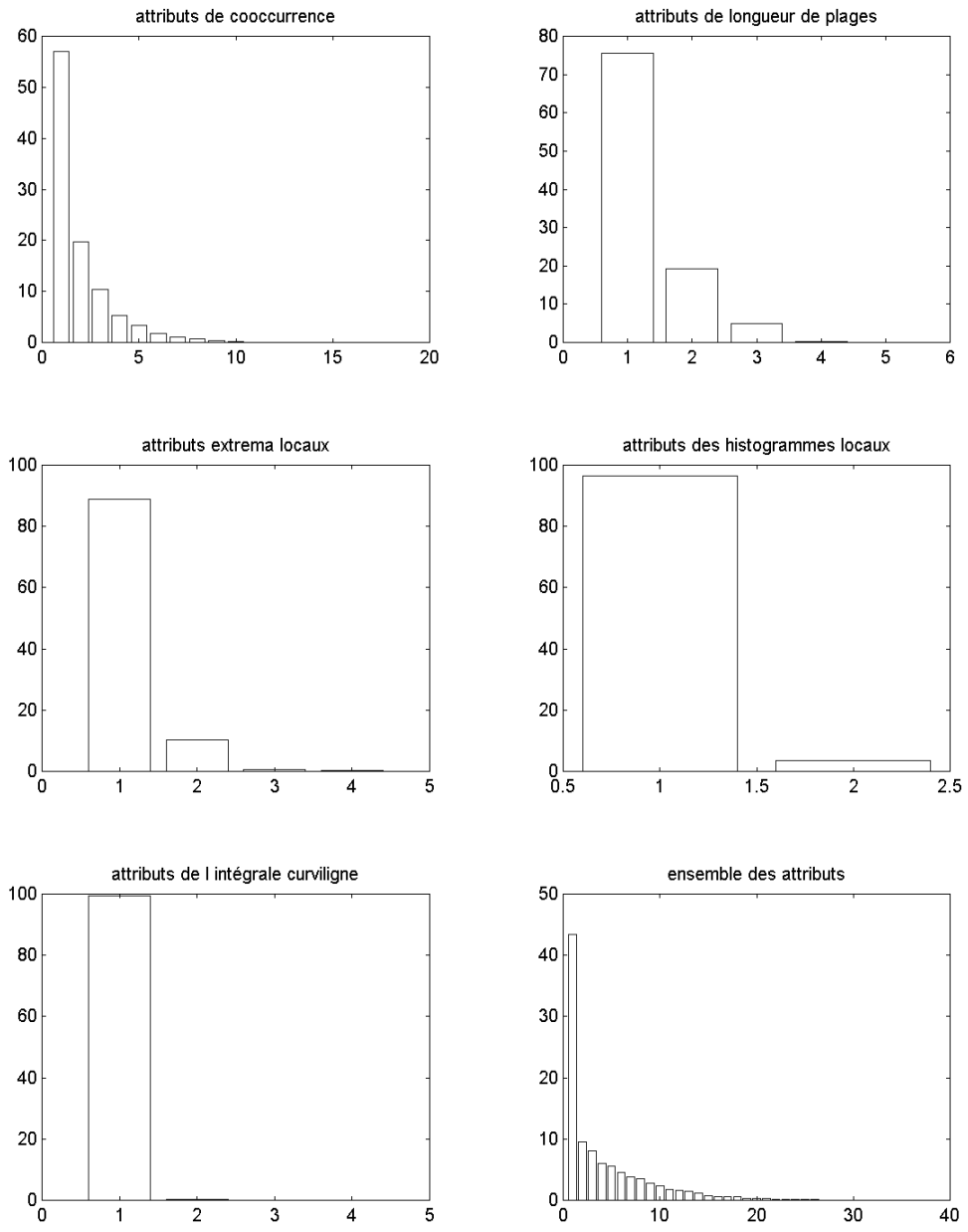


FIG. 2.13 – *Apport de chaque valeur propre pour chaque type d'attribut en pourcentage classé dans l'ordre décroissant*

rapport à l'attribut $\mathbf{a}^j$ si et seulement si :

$$|Correl(\mathbf{a}^i,\mathbf{a}^j)| > S$$

La valeur S correspond à la redondance maximale tolérée pour deux attributs. En pratique, ce nombre est choisi relativement proche de 1 ceci pour ne pas perdre d'information utile pour l'analyse d'une texture.

Une fois que l'on sait que deux attributs sont redondants, il reste à choisir celui que l'on va écarter. Pour ce faire, nous définissons le critère suivant en exploitant la redondance d'un attribut avec tous les autres :

$$\begin{cases} si\ \sum_k Correl(\mathbf{a}_k,\mathbf{a}_i) > \sum_k Correl(\mathbf{a}_k,\mathbf{a}_j)\ alors\ l'attribut\ \mathbf{a}_i\ est\ \text{écarté} \\ \\ sinon\ l'attribut\ \mathbf{a}_j\ est\ \text{écarté} \end{cases}$$

A ce stade, il ne reste plus que des attributs ayant une redondance admissible réglée par la valeur du seuil S et satisfaisant le critère global de redondance. Nous avons essayé de quantifier le nombre d'attributs non redondants à partir de la matrice de corrélation pour un seuil donné (cf. TAB. 2.4). Ceci est réalisé dans un premier temps pour chaque type d'attribut. Concernant la réduction des paramètres de différents types, nous avons utilisé le même critère. Ainsi, pour un seuil de 0.95, nous obtenons 19 attributs sur les 30 initiaux. Il faut noter que l'attribut COOC14 a été éliminé puisqu'il est fortement corrélé avec LDP5 (coefficient de corrélation égale à 1) alors qu'il apparaissait parmi les attributs de la matrice de cooccurrence ayant un facteur de corrélation inférieur à 0.95.

Nous avons utilisé la méthode CHAVL [LP85] (cf. section 1.2.2.2.b) pour analyser la redondance des attributs de texture classiques (cf FIG. 2.14). L'indice d'agrégation calculé dans la méthode CHAVL est basé sur la vraisemblance de liens entre deux attributs calculée à partir du facteur de corrélation. L'arbre obtenu permet de confirmer les résultats obtenus par sélection des attributs non redondants. En effet, la redondance d'attributs de même type et de types différents est mise en évidence en considérant les premiers niveaux d'agrégations. On peut constater par exemple que les attributs de l'intégrale curviligne ont été tous regroupés dans un sous-arbre montrant leur redondance relativement importante. Cet arbre permet aussi de mettre en évidence la forte corrélation de deux attributs de types différents. Les attributs *cooc*14 et *ldp*5 ont été regroupés au premier niveau comme nous pouvions l'attendre puisque la valeur absolue de leur facteur de corrélation vaut 1.

2.3.1.3 Classement de textures

Afin d'apprécier le comportement de ces descripteurs avant et après réduction, nous avons essayé de classer les textures de la mosaïque par ordre de similarité [CR99]. Le principe de classement est le suivant :

- chaque texture T_i, $i = 1,..,50$ est décrite par un vecteur d'attributs $\mathbf{a}^i$,

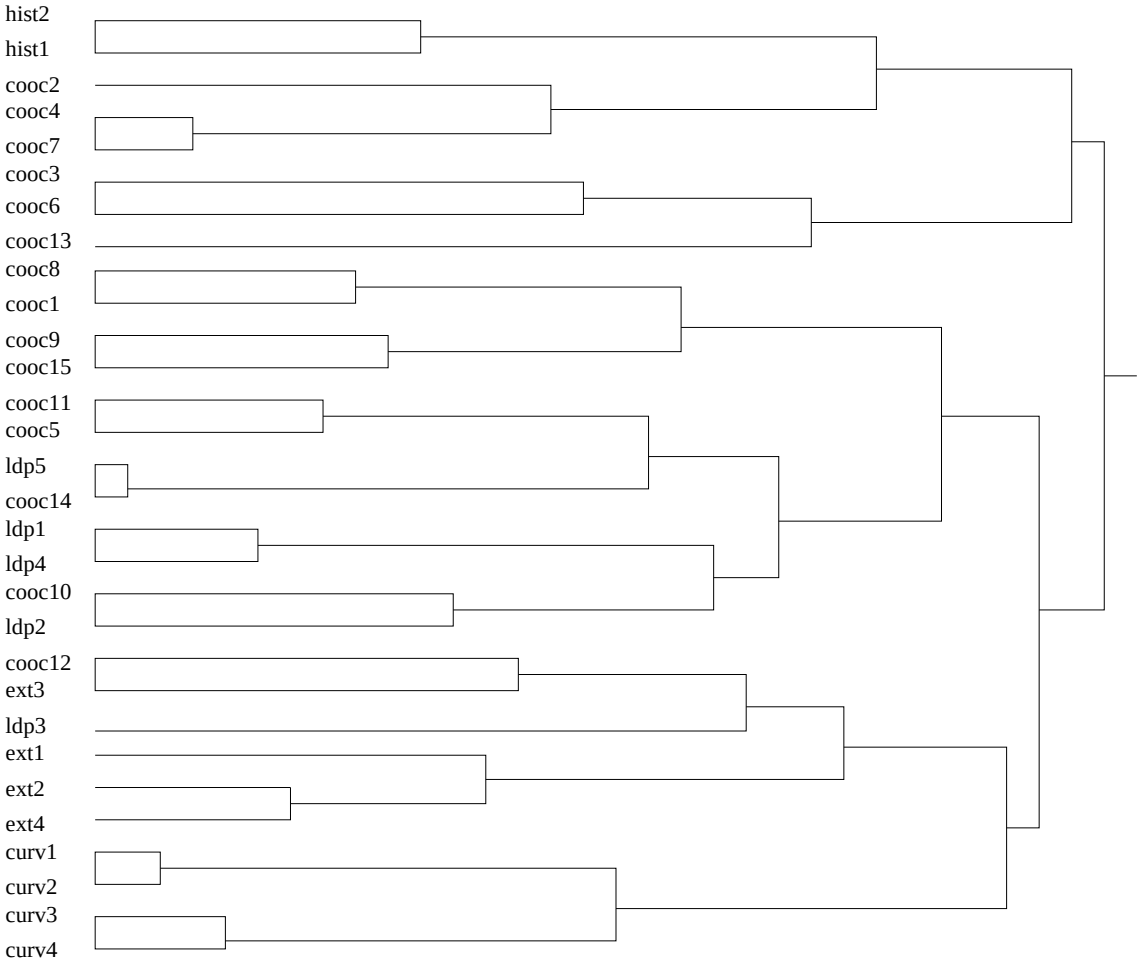


FIG. 2.14 – *Classification des attributs de texture classiques*

type d'attribut	seuil	nombre	attributs non corrélés
cooccurrence	0.95	9/15	COOC2, COOC3, COOC6, COOC7, COOC9, COOC12, COOC13, COOC14, COOC15
cooccurrence	0.90	6/15	COOC2, COOC3, COOC6, COOC9, COOC12, COOC13
cooccurrence	0.55	4/15	COOC2, COOC3, COOC9, COOC13
longueur de plages	0.95	3/5	LDP2, LDP3, LDP5
longueur de plages	0.65	2/5	LDP3, LDP5
extrema locaux	0.95	4/4	EXT0, EXT45, EXT90, EXT135
extrema locaux	0.80	3/4	EXT0, EXT90, EXT135
extrema locaux	0.55	2/4	EXT0, EXT90
intégrale curviligne	0.95	2/4	CURV0, CURV135
intégrale curviligne	0.90	1/4	CURV0
histogrammes locaux	0.95	2/2	HIST1, HIST2
histogrammes locaux	0.80	1/2	HIST1
tous	0.95	19/30	COOC2, COOC3, COOC5, COOC6, COOC7, COOC9, COOC12, COOC13, COOC15, LDP3, LDP5, HIST1, HIST2, EXT0, EXT45, EXT90,EXT135, CURV0, CURV135
tous	0.90	15/30	COOC2, COOC3, COOC5, COOC6, COOC12, COOC13, LDP3 LDP5, HIST1, HIST2, EXT0, EXT45, EXT90, EXT135, CURV0
tous	0.60	9/30	COOC2, COOC3, COOC6, COOC13, COOC14, LDP5, HIST1 EXT0, EXT135
tous	0.50	6/30	COOC2, COOC3, COOC14, HIST1, EXT135, CURV0

TAB. 2.4 – *Nombre d'attributs faiblement corrélés calculé à partir de la matrice de corrélation pour une valeur du seuil de corrélation fixé*

- on détermine dans un premier temps les textures les plus dissemblantes parmi les 50 en utilisant la distance euclidienne (notés T^1 et T^{50}),
- on détermine ensuite la texture dont le vecteur d'attributs est le plus proche de celui de T^1 (resp. T^{50}) que l'on note $T^1(1)$ (resp. $T^{50}(1)$). Si la distance entre T^1 et $T^1(1)$ est plus faible que celle entre T^{50} et $T^{50}(1)$ alors la texture $T^1(1)$ devient T^2 sinon $T^{50}(1)$ devient T^{49}. On réitère le procédé en cherchant la texture dont l'attribut est le plus proche de celui de T^2 ou de T^{49} et ainsi de suite.

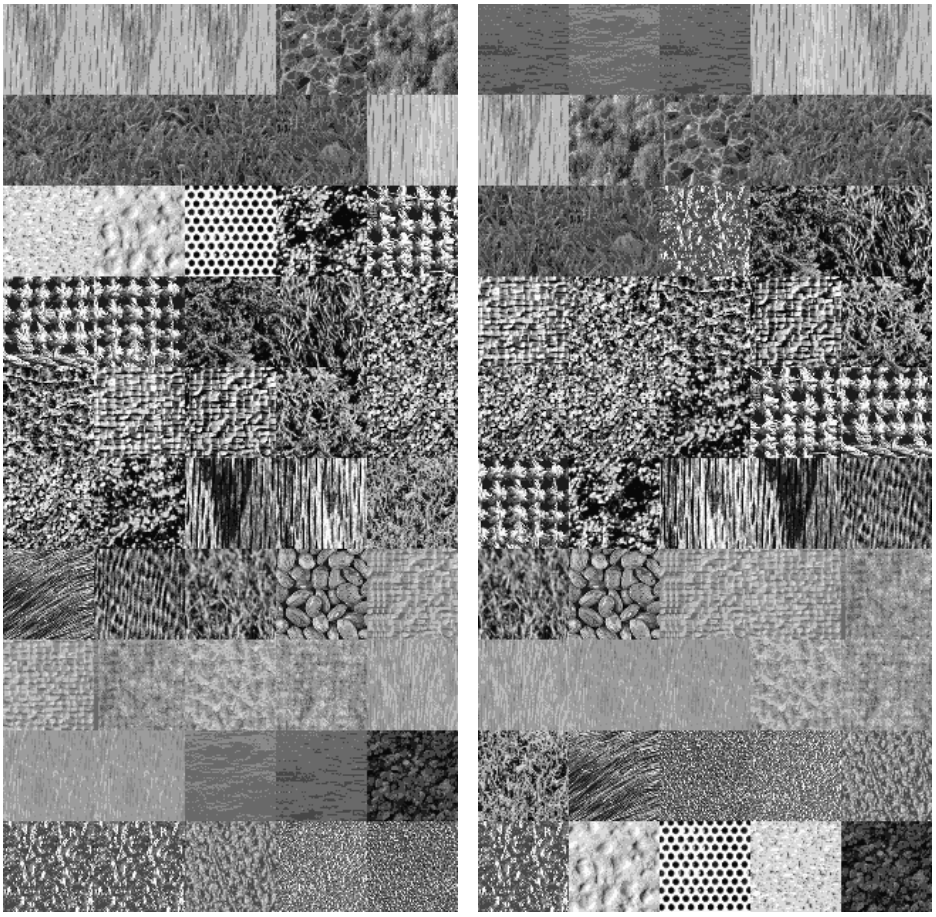
a) Avant réduction du nombre d'attributs

On obtient une mosaïque de textures classées par ordre de similarité ligne par ligne. Le résultat de classement est consigné sur la figure FIG 2.15(a).

Ce qui est important de constater dans ce résultat est que deux textures visuellement similaires soient classées côte à côte. Il peut arriver que deux textures relativement différentes aient un classement proche. Ceci vient, d'une part, de notre méthode de classement (la place d'une texture n'est pas forcément liée à la place de la texture la plus proche mais à celles des textures déjà classées) et, d'autre part, de la non pertinence des paramètres (difficulté de discrimination de deux textures différentes à partir des descripteurs utilisés).

b) Après réduction du nombre d'attributs

Nous présentons le résultat de classement des textures de la mosaïque de la figure FIG. 2.12 en sélectionnant les attributs non redondants (cf. FIG. 2.15(b)). Nous avons pris un seuil de 95% permettant d'obtenir uniquement 19 attributs à partir des 30 initiaux. Le résultat obtenu est cohérent et est équivalent à celui obtenu en utilisant tous les attributs.



(a) avant sélection

(b) après sélection avec un seuil de 95%

FIG. 2.15 – *Résultat de classement des textures de la mosaïque à partir de l'ensemble des descripteurs de texture avant et après sélection des attributs non corrélés*

2.3.1.4 Classification de textures

Nous avons réalisé une classification supervisée en utilisant un critère de minimisation de la distance euclidienne. Chaque classe de textures dans la mosaïque est caractérisée un vecteur d'attributs servant de référence. La matrice de confusion est

donnée dans le tableau TAB. 2.3.1.4. Le taux de bonne classification en utilisant tous les attributs de texture est de 75.8%. En utilisant les attributs après réduction (un vecteur d'attributs de dimension 19 au lieu de 30), le taux de bonne classification obtenu est légèrement meilleur car il est de 76.6% (cf. TAB. 2.3.1.4).

2.3.1.5 Conclusions

Cette étude nous a permis d'évaluer la redondance entre différents attributs. Le premier point de cette étude concerne la redondance des descripteurs de texture du même type et nous a permis de dégager plusieurs enseignements :

- les attributs de la matrice de cooccurrence sont relativement redondants. En effet, seuls 9 des 15 d'entre eux ont été conservés,
- les attributs de la matrice de longueur de plages sont un peu plus redondants que ceux de la matrice de cooccurrence et sont moins nombreux (3 attributs sur 5 sont conservés).
- les deux attributs des histogrammes locaux apportent une information complémentaire. Ils sont donc à conserver,
- les attributs de l'intégrale curviligne montrent une forte corrélation pour différentes orientations d'analyse. Deux attributs sur quatre sont conservés,
- les attributs des extrema locaux apportent une information complémentaire si l'on considère deux orientations orthogonales.

Le second point concerne la redondance entre attributs de différents types. Nous avons noté néanmoins une relative redondance entre les attributs de cooccurrence et de longueur de plages. Ceci peut s'expliquer par le fait que l'information extraite est relativement proche. Par ailleurs, les autres types d'attributs sont relativement moins corrélés.

Nous avons montré l'intérêt de combiner ces différents attributs et n'utiliser qu'un sous-ensemble étant donné un seuil de corrélation admissible.

Ces attributs permettent une bonne description de la texture rencontrée mais ne donnent aucune information perceptuelle sur le caractère déterministe d'une texture ou sur sa finesse. Le caractère déterministe d'une texture donne une indication sur le type de la texture (texture structurelle, structurelle-statistique, statistique). La finesse d'une texture, quant à elle, donne une information sur la taille du motif de la texture considérée (microscopique, macroscopique) et donc peut permettre d'adapter la taille de la fenêtre d'analyse.

2.3.2 Définition de nouveaux attributs de texture

Nous passons tout d'abord en revue les différentes définitions de la texture afin de mettre en évidence leurs caractéristiques essentielles que nous tenterons d'exploiter

	d002	d004	d009	d019	d023	d024	d028	d029	d032	d037	d038	d053	d057	d066	d068	d070	d074	d077	d084	d093	d112
d002	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d004	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d009	0.00	12.50	0.00	0.00	0.00	0.00	12.50	12.50	25.00	0.00	12.50	12.50	0.00	0.00	0.00	0.00	0.00	0.00	12.50	0.00	0.00
d019	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d023	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d024	0.00	25.00	0.00	0.00	0.00	25.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	50.00	0.00	0.00
d028	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d029	0.00	50.00	0.00	0.00	0.00	50.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d032	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d037	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	66.67	33.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d038	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d053	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d057	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d066	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d068	0.00	0.00	0.00	0.00	20.00	0.00	0.00	20.00	0.00	0.00	0.00	0.00	40.00	0.00	0.00	0.00	0.00	20.00	0.00	0.00	0.00
d070	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00
d074	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00
d077	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00
d084	50.00	0.00	0.00	0.00	0.00	0.00	0.00	50.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d093	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
d112	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00

TAB. 2.5 – *Matrice de confusion en utilisant l'ensemble des attributs de texture : taux de bonne classification* 75.8%

	d002	d004	d009	d019	d023	d024	d028	d029	d032	d037	d038	d053	d057	d066	d068	d070	d074	d077	d084	d093	d112
d002	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d004	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d009	0.00	12.50	0.00	0.00	0.00	0.00	12.50	12.50	25.00	0.00	12.50	12.50	0.00	0.00	0.00	0.00	0.00	0.00	12.50	0.00	0.00
d019	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d023	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d024	0.00	0.00	0.00	0.00	0.00	25.00	0.00	25.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	50.00	0.00	0.00
d028	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d029	0.00	33.33	0.00	0.00	0.00	33.33	0.00	16.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	16.67	0.00	0.00	0.00
d032	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d037	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	66.67	33.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d038	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d053	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d057	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d066	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d068	0.00	0.00	0.00	0.00	20.00	0.00	0.00	20.00	0.00	0.00	0.00	0.00	40.00	0.00	0.00	0.00	0.00	20.00	0.00	0.00	0.00
d070	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00
d074	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00
d077	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00
d084	25.00	0.00	0.00	25.00	0.00	0.00	0.00	50.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d093	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
d112	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00

TAB. 2.6 – *Matrice de confusion après réduction du nombre d'attributs : taux de bonne classification* 76.6%

en proposant de nouveaux attributs.

2.3.2.1 Définition de la texture

Le traitement d'images naturelles met en évidence la nécessité de prendre en compte pour une interprétation postérieure, des régions non uniformes au sens du niveau de gris (voir FIG. 2.16). Ces régions sont dites texturées. Pour définir une texture, nous distinguons deux approches :

- l'approche perceptuelle : une texture est décrite par des propriétés visuelles ;
- l'approche formelle : une texture est définie à partir des primitives la composant et en intégrant leurs interactions mutuelles.

a) Approche perceptuelle

La notion de texture est abordée ici au travers de sa perception. Cette approche consiste à décrire une texture à partir de propriétés perceptuelles telles que la complexité ou la granularité.

Le mot texture (du latin *textura*) a tout d'abord désigné l'arrangement des fils d'un matériau tissé. A partir du VI ème siècle, la texture définit plus généralement l'arrangement des éléments d'une matière, elle caractérise sa structure. Une texture représente une variation d'un ou plusieurs paramètres caractéristiques d'un objet, à une petite échelle devant la taille de ce dernier. La prise en compte de la taille est importante : les fils d'un tissu définissent la texture d'un vêtement à l'échelle humaine, mais en effectuant un zoom sur ce dernier, le fil devient le matériau dont les fibres forment la texture.

Unser [Uns84] définit la texture comme suit :
"Une texture est une région d'une image pour laquelle il est possible de définir une fenêtre de dimensions minimales, telle qu'une observation au travers de celle-ci se traduit par une perception (impression) visuelle identique pour toutes les translations possibles de cette fenêtre à l'intérieur de la région considérée".

Cette propriété d'invariance par translation de perception visuelle d'une texture est exploitée dans la plupart des méthodes de classification à partir d'attributs de texture [Uns95], [CK93], [HB94], [LF93].

Selon Picard [Pic96], il est difficile d'établir une définition de la texture. Il est cependant possible de dégager trois propriétés essentielles :

- la complexité d'une texture est difficilement quantifiable,
- une texture est caractérisée par des variations importantes dans les hautes fréquences. Une zone uniforme ou quasiment lisse n'est pas une texture pour Picard,

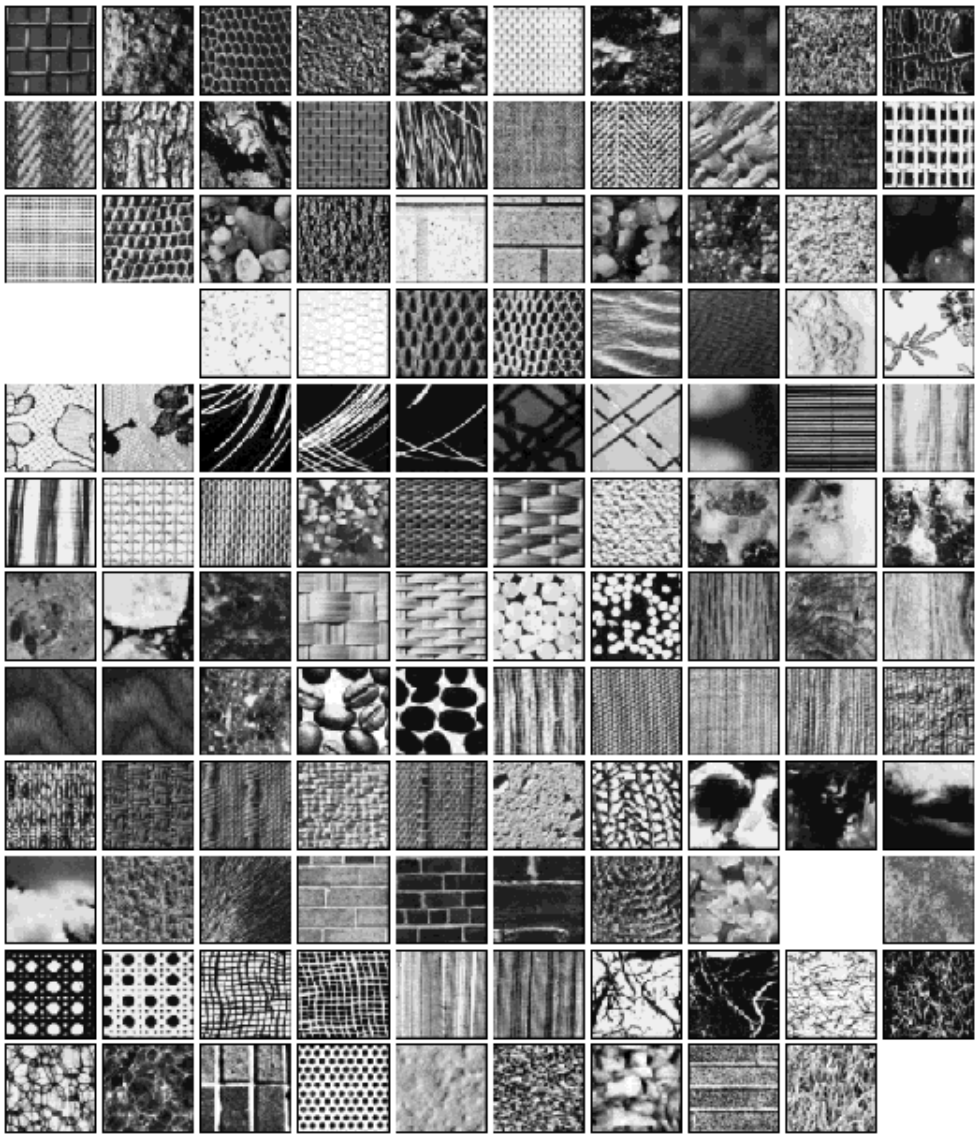


FIG. 2.16 – *Présentation des différentes textures de l'album de Brodatz*

- la notion de texture n'est valable que pour un ensemble de résolutions d'observation donné.

Selon Zhang [Zha96a], la notion de texture découle de la nécessité de décrire des informations d'une image de la manière la plus simple possible : *"Une texture peut être considérée comme une façon de décrire les corrélations entre les différentes zones d'une région de l'image"*.

Notons que cette simplicité de description s'accompagne généralement de l'emploi de termes 'vagues' dont la complexité est difficilement quantifiable. La texture permet de décrire des zones complexes de l'image composées de motifs élémentaires plus ou moins faciles à déterminer.

En résumé, les principales propriétés visuelles d'une texture que l'on dégage de la littérature sont les suivantes :

- Une texture ne peut être définie que sur un voisinage dont la taille dépend de la taille des motifs qui la composent.
- La notion de texture est liée à la résolution de l'observation. En effet, une région peut apparaître texturée à une certaine échelle et uniforme à une autre.

Ces propriétés visuelles sont importantes lors de la caractérisation d'une texture. Plusieurs paramètres de texture découlant de ces propriétés (granularité, caractère déterministe) permettent de discriminer différentes textures.

b) Approche formelle

L'approche formelle consiste à déterminer les éléments fondamentaux composant une texture et à décrire leurs interactions.

La définition du dictionnaire indique que la texture est la répétition spatiale d'un motif de base dans diverses directions de l'espace.

Selon Julesz [Jul75], la texture apparaît comme la répartition spatiale régulière d'un unique motif géométrique appelé "texton". Il faut noter que cette définition n'est valable que pour les textures déterministes.

Selon Haralick [Har79], une texture peut être décomposée suivant deux dimensions. La première concerne la description des primitives tonales qui la composent et la seconde définit les interactions entre ces propriétés tonales. On peut donc, selon ce schéma discerner trois types de texture :

- textures macroscopiques ou structurelles : ces textures présentent une structure spatiale régulière pouvant être générée par la répétition d'un motif ou d'une primitive. Elles représentent en général un milieu artificiel créé par l'homme (par exemple un mur de briques), elles peuvent également correspondre à des

images de synthèse,

- textures microscopiques ou aléatoires : la texture est prise en compte comme la réalisation d'un processus stochastique bidimensionnel où il est difficile d'isoler des motifs élémentaires. Le seul motif que l'on puisse trouver dans une texture granit, par exemple, est un grain d'une couleur donnée et sa répartition dans l'image est aléatoire,
- textures hybrides : l'interaction entre les primitives tonales de ce type de textures est aléatoire. Une photo aérienne d'une forêt est composée de textures composites où un arbre peut être assimilé à une primitive et son arrangement est purement aléatoire.

Francos et al [FMP93] décomposent les textures en trois classes en s'appuyant sur la décomposition de Wold. La décomposition de Wold d'un champ aléatoire 2-D est une superposition de trois composantes orthogonales :

- un champ purement aléatoire ;
- un champ généralisé évanescent ;
- un champ harmonique.

Cette décomposition dite de Wold permet de modéliser le caractère déterministe ou aléatoire d'une texture. Cette décomposition ne prend pas en compte la notion de motif de la texture considérée.

c) Définition du motif d'une texture

L'approche formelle tente de définir les composantes d'une texture ainsi que leurs interactions. Toutes les définitions de ce type s'accordent à dire qu'une texture est la répétition spatiale d'un motif. La notion de motif donnée par Haralick [Har79] et Francos [FMP93] reste assez floue, c'est ce que nous nous proposons de préciser par la suite. Nous définissons le motif d'une texture en prenant en compte trois composantes :

- le motif de base : c'est la plus petite structure spatiale de la texture ,
- la transformation du motif de base : elle caractérise d'éventuelles déformations géométriques et des modifications des niveaux de gris du motif au sein de la même texture,
- la loi de probabilité de la transformation du motif : elle permet de définir la transformation d'un motif donné.

Nous définissons donc le motif ou la primitive d'une texture de la façon suivante : *"Le motif d'une texture est la plus petite structure spatiale contenue dans une texture dont la forme est régie par la même loi de probabilité au sein d'une texture".*

Prenons l'exemple d'une texture gaussienne microscopique (de motif égal à un pixel). Dans ce cas simple, la loi de probabilité décrivant le motif de la texture est une gaussienne dont les paramètres peuvent être obtenus en analysant

l'histogramme des niveaux de gris.

De même, l'interaction entre ces motifs décrit la fréquence d'apparition du motif ainsi que sa localisation. La stabilité de la loi de probabilité concernant le motif et les interactions entre les motifs est pour nous la propriété fondamentale dans la définition d'une texture.

Nous distinguons trois types de texture en caractérisant à la fois les motifs et leur interaction :

- texture déterministe ou structurelle : le motif spatial de la texture est déterministe et est répété suivant une règle de déplacement déterministe. Un exemple d'une telle texture est un mur composé de briques identiques uniformément disposées.
- texture statistique : le motif est disposé de façon aléatoire au sein de la même texture. Une texture de granit par exemple est composée d'un grain de niveau de gris aléatoire répété de façon aléatoire.
- texture structurelle-statistique : il en existe deux types :
  - texture composée d'un motif déterministe dont le positionnement est aléatoire : une image contenant un amas de grains de café est un exemple d'une telle texture,
  - texture composée d'un motif aléatoire dont le positionnement est déterministe : une image aérienne représentant un parking de voitures est un exemple d'une telle texture. La règle de positionnement des voitures est fixe alors que le type de voiture dans une place du parking varie.

Le caractère déterministe ou aléatoire est la propriété prépondérante dans cette définition de la texture. De plus, il est possible de décrire la taille du motif composant la texture. Cette information sur la finesse d'une texture peut se révéler être très utile notamment pour le choix du support de modélisation ou d'analyse d'une texture.

On peut remarquer qu'une zone uniforme peut être prise en compte comme une texture à la vue de cette définition. En effet, le motif d'une région uniforme est le pixel et la règle de déplacement est la translation dans toutes les directions. Par la suite, on distingue néanmoins une zone uniforme d'une zone texturée car la plupart des techniques de traitement d'images ne sont pas adaptées aux deux cas. En effet, le contenu informationnel d'une texture est très supérieur à celui d'une région uniforme, donc plus difficile à traiter.

Après avoir définie la texture, nous allons déterminer à partir de la fonction d'autocorrélation un indice permettant d'identifier le caractère déterministe ou aléatoire d'une texture (texture structurelle, structurelle-statistique, statistique) et un paramètre lié à la finesse ou au motif la constituant (texture macroscopique,

microscopique).

L'ensemble de ces informations doit être invariant par rotation. En effet, le caractère déterministe ou la finesse d'une texture ne doit pas dépendre de son orientation.

La fonction d'autocorrélation permet d'obtenir de multiples informations perceptuelles sur la texture. Nous allons exploiter la fonction d'autocorrélation normalisée sur une image centrée (de moyenne nulle) qui correspond à la fonction d'autocovariance normalisée.

Pour extraire ces informations perceptuelles, nous allons modéliser la fonction d'autocovariance normalisée 1D d'une texture [RCCO99].

2.3.2.2 Modélisation de la fonction d'autocorrélation 1D

Nous présentons un modèle d'une fonction d'autocorrélation 1D s'appuyant sur le formalisme mathématique de Francos et *al* [FMP93].

Dans la décomposition de Wold d'un champ aléatoire 1D, il est fait l'hypothèse de la stationnarité du signal 1D. Un champ est dit homogène ou stationnaire au sens large si la moyenne et la variance du signal sont des constantes. Cette condition est presque indispensable en segmentation d'images.

Un champ harmonique 1D I_1 peut être modélisé par une somme de p sinusoïdes :

$$I_1(r) = \sum_p \gamma_p sin(2\pi f_p r + \phi_p)$$

où γ_p est l'amplitude de la $p^{\text{ème}}$ composante sinusoïdale du champ déterministe, f_p la fréquence associée et ϕ_p la phase.

La fonction d'autocovariance d'un champ harmonique 1D I_1 s'écrit sous la forme :

$$FAC_{I_1}(r) = \gamma_0^2 + 2\sum_{p>0} \gamma_p^2 cos(2\pi f_p r + \phi_p)$$

De même, la fonction d'autocovariance d'un champ purement aléatoire I_2 est une fonction qui peut s'exprimer de la façon suivante [FMP93] :

$$FAC_{I_2}(r) = e^{-\alpha r}$$

Un champ évanescent 1D I_3 peut s'écrire sous la forme suivante [FMP93]:

$$I_3(r) = b(r).\sum_p \gamma_p sin(f_p r + \phi_p)$$

où b est un processus aléatoire 1D. La fonction d'autocovariance associée est de la forme :

$$FAC_{I_3}(r) = e^{-\alpha_1 r} + e^{-\alpha_2 r} \sum_p \gamma_p cos(2\pi f_p r + \phi_p)$$

La fonction d'autocovariance d'un champ évanescent est caractérisée par une décroissance rapide pour des distances de déplacement faibles et par des oscillations d'amplitudes atténuées.

Ainsi, la fonction d'autocorrélation ou d'autocovariance d'un champ aléatoire 1D I peut être modélisée comme suit :

$$F\hat{A}C_I(r) = e^{-\alpha r} + \gamma.e^{-\beta r} cos(2\pi f r + \phi) + \delta + \epsilon(r) \tag{2.2}$$

Le terme $\epsilon(r)$ correspond à l'erreur de modélisation de la fonction d'autocovariance normalisée. Dans le cas d'un champ purement aléatoire, le premier terme de cette expression doit être prépondérant, ce qui est réalisé à partir d'une valeur de α très faible et une valeur de β élevée ou γ faible. Dans le cas déterministe, le coefficient β doit être proche de 0. Enfin, le cas évanescent intermédiaire est caractérisé par une valeur de β positive moyenne pour modéliser les oscillations amorties présentes dans la fonction d'autocovariance normalisée.

La taille du motif M d'une texture T peut être estimée à partir des paramètres du modèle. Dans le cas déterministe, la taille du motif est approchée par la période $1/f$. Dans le cas hybride ou aléatoire, la dimension M_T du motif est approximée en considérant la valeur de M_T pour laquelle $\hat{FAC}(M_T)$ est négligeable.

2.3.2.3 Calcul de la fonction d'autocovariance normalisée 1D d'une texture

Nous présentons une méthode de calcul de la fonction d'autocovariance normalisée 1D d'une texture à partir de celle en 2D. On note F cette fonction et elle se calcule de la façon suivante:

$$F(r) = \frac{1}{\pi r} \sum_{(i,j)\in\hat{C}_r} F\tilde{A}C(i,j) \quad \forall r > 0$$

où $\hat{C}_r = \{(i,j)/i \geq 0, \sqrt{i^2+j^2} = r\}$ est l'ensemble des points sur le demi-cercle de rayon r (cf. FIG. 2.17) et $F\tilde{A}C$ est définie de la façon suivante :

$$F\tilde{A}C(i,j) = \begin{cases} FAC(i,j) & si\ m = \sqrt{i^2+j^2} \in \mathbb{N} \\ \sum_{k=1}^{4} \frac{d_k.FAC(s_x^{(k)},s_y^{(k)})}{d_1+d_2+d_3+d_4} & sinon \end{cases} \tag{2.3}$$

Le premier terme est la fonction d'autocovariance normalisée 2D définie de la façon suivante :

$$FAC(i,j) = \frac{1/NW \sum_{s=(k,l)\in W} T(s)T(k+i,l+j)}{1/N \sum_{k,l} T^2(k,l)}$$

où NW et W sont respectivement le nombre de points et la région sur laquelle le produit $I(s)I(k+i,l+j)$ est calculé, N le nombre total de pixels, T est la fonction de luminance de moyenne nulle de la texture considérée, i et j sont les déplacements horizontaux et verticaux.

Le second terme de la fonction $F\tilde{A}C$ correspond à l'interpolation de la fonction d'autocovariance au point m sur le demi cercle $\hat{C}_r$. Le terme d_k correspond à la distance euclidienne $d(s^{(k)},m)$ entre le point m et le point $s^{(k)} = (s_x^{(k)},s_y^{(k)})$.

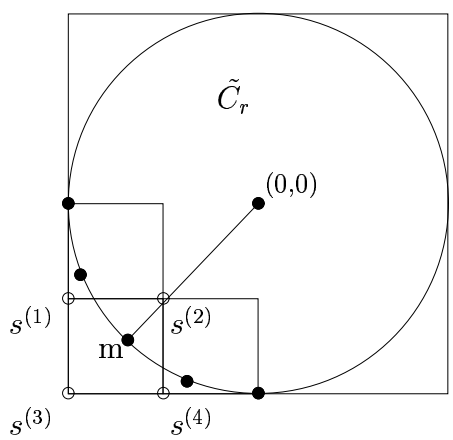


FIG. 2.17 – *Interpolation d'un point sur le demi-cercle* $\hat{C}_r$

Soit $F(r)$ la valeur traduisant la corrélation moyenne d'un pixel avec un autre distant de r dans toutes les directions, cette fonction s'apparente à la fonction d'autocorrélation d'un signal 1D. En effet, il est facile de montrer qu'elle est invariante par translation de l'image (la fonction F est calculée à partir de l'autocovariance 2D qui a cette propriété). La fonction F est aussi invariante par rotation. Cette propriété est obtenue en prenant en compte la corrélation des pixels dans toutes les directions.

2.3.2.4 Détermination des paramètres du modèle

La fonction F calculée précédemment est analysée comme la fonction d'autocorrélation d'un signal 1D. Les paramètres de modélisation de cette fonction sont

obtenus par moindres carrés, c'est à dire en minimisant la fonctionnelle suivante :

$$\sum_{i=1}^{r_q}(F(r_i) - \tilde{F}(r_i))^2$$

où r_q est le rayon maximal considéré et $\tilde{F}$ le modèle de la fonction F.

Afin de déterminer les paramètres minimisant cette fonctionnelle, nous avons utilisé un algorithme génétique [HJK95b] (GAOT : Genetic Algorithm for Optimization Toolbox). Cette approche est particulièrement adaptée à notre problème puisque la fonction à minimiser est non convexe (à cause de la présence du cosinus dans le modèle). D'autres approches telles que le gradient conjugué ou le recuit simulé ne donneraient que des extrema locaux dans la plupart des cas [HJK95a].

Les algorithmes génétiques déterminent les solutions d'une fonction en simulant l'évolution d'une population jusqu'à la survie des individus les plus adaptés. Le choix d'une population initiale importante garantit la convergence vers un minimum global. Les survivants sont des individus obtenus par croisement, mutation ou sélection d'individus de la génération précédente.

2.3.2.5 Caractère déterministe d'une texture

Nous obtenons un vecteur de paramètres $(\alpha,\beta,\gamma,f,\phi,\delta,\epsilon)$ que nous allons exploiter pour obtenir des propriétés perceptuelles d'une texture comme le caractère déterministe. Notre modèle peut se décomposer en deux termes : le premier concerne la composante aléatoire de la texture et le deuxième la composante harmonique.

Nous définissons le caractère déterministe Δ_T d'une texture T en considérant les coefficients β et γ du modèle proposé. Dans le cas d'une texture aléatoire, le second terme du modèle doit être négligeable ce qui est réalisé pour une valeur de β élevée ou une valeur de γ très faible. Tandis que dans le cas purement déterministe, cette composante est une sinusoïde d'amplitude constante, correspondant à une valeur de β nulle ou assimilable et à une valeur de γ strictement positive. Le cas intermédiaire (structurel-statistique) correspond à une valeur de β moyenne réglant l'atténuation de la sinusoïde du second terme du modèle.

L'indice Δ_T exprimant le caractère déterministe d'une texture T s'écrit sous la forme :

$$\Delta_T = \begin{cases} 0 & si\ |\tilde{\gamma}| \approx 0 \\ 1 & si\ \tilde{\beta} \approx 0 \\ 2|\tilde{\gamma}|e^{-\tilde{\beta}} & sinon \end{cases}$$

où $\tilde{\beta}$ et $\tilde{\gamma}$ sont les valeurs normalisées des paramètres (β,γ). En effet, dans l'étape d'estimation des paramètres, la valeur β doit être comprise entre 0 et 1, et

γ entre -0.5 et 0.5.

Le caractère déterministe d'une texture correspond à des valeurs de Δ_T comprises entre 0 et 1. Plus la valeur de Δ_T est proche de 1, plus la texture est déterministe et plus la valeur de Δ_T est proche de 0, plus elle est aléatoire.

Il est possible d'affiner cette caractérisation d'une texture en considérant la finesse d'une texture. Cette information, primordiale en segmentation puisqu'elle conditionne la taille de la fenêtre d'analyse d'une texture, est aussi obtenue à partir des paramètres du modèle précédent.

2.3.2.6 Finesse d'une texture

La finesse d'une texture est une propriété perceptuelle importante puisqu'elle caractérise la taille du motif d'une texture.

Selon Gagalowicz [Gag83], une texture fine est caractérisée par une fonction d'autocorrélation tendant vers 0 rapidement en fonction de la distance de déplacement. Cette même fonction pour une texture grossière (primitives de grande taille) décroît beaucoup plus lentement. Dans le cas des textures déterministes, cette analyse ne doit se faire que sur la période de la texture considérée.

Cette définition caractérise la variation de la corrélation entre pixels plus ou moins proches pour une texture donnée.

Nous définissons la finesse d'une texture comme la caractérisation de la taille du motif élémentaire composant une texture à une résolution donnée. La résolution est liée à la distance d'observation d'une texture ou à la taille de la fenêtre d'analyse de celle-ci. La finesse contrairement au caractère déterministe ou aléatoire d'une texture, varie en fonction de la résolution de l'observation.

La finesse d'une texture peut être décrite à partir de l'évolution de l'information contenue dans la fonction d'autocorrélation et en particulier dans la fonction F. L'attribut que nous proposons d'extraire de cette fonction traduit la taille minimale d'analyse de la texture considérée. Dans le cas déterministe, cette taille s'apparente à la période de la texture. Dans le cas contraire, elle est estimée à partir de la fonction F.

La décision d'appartenance à la classe déterministe a été réalisée en considérant l'indice Δ_T de la texture. Dans le cas où cet indice est supérieur à 0.5 (seuil obtenu après expérimentations sur un ensemble significatif d'images), nous considérons que la texture est déterministe. Dans ce cas, la période de la texture est estimée en considérant la plus grande période du signal 2D. Il peut arriver en effet qu'une

texture déterministe soit composée de plusieurs composantes harmoniques.

Dans le cas non purement déterministe, la finesse M_T d'une texture T est obtenue de la façon suivante (cf. FIG. 2.18) :

$$M_T = arg \min_{r>0} F(r) < \delta$$

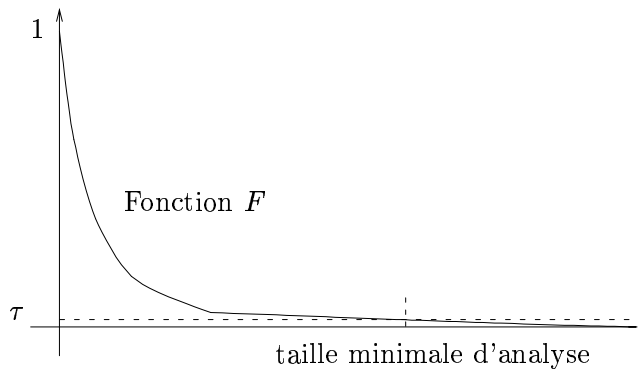


FIG. 2.18 – *Détermination de la finesse d'une texture : cas non déterministe*

Cette formulation de la finesse d'une texture présente deux avantages. D'une part, la définition de la finesse d'une texture, comme étant la caractérisation de la taille minimale du support d'analyse de celle-ci, a un sens dans le cas déterministe ou aléatoire. D'autre part, cette information est immédiatement exploitable dans la phase de segmentation.

Plus une texture est fine (resp. grossière), plus la taille du voisinage d'analyse de celle-ci est petite (resp. grande).

2.3.2.7 Résultats expérimentaux

Les résultats expérimentaux se décomposent en trois étapes :
- la première étape consiste à valider le modèle de texture et l'estimation de ses paramètres,
- la deuxième étape concerne l'étude de la corrélation de ces paramètres avec les descripteurs de texture précédents,
- la dernière étape porte sur l'évaluation de l'apport des paramètres du modèle dans une phase d'analyse de textures.

a) Validation du modèle de texture

Nous avons utilisé la mosaïque de 50 textures de Brodatz de la figure FIG.2.12 et un ensemble de 12 textures 256×256 pixels (cf. FIG.2.19), soit 62 images

(certaines textures apparaissent pour les deux tailles d'image).

Nous avons calculé la fonction d'autocovariance normalisée pour chaque texture de cet ensemble ainsi que la fonction $r \mapsto F(r)$ associée. Nous avons déterminé les paramètres du modèle de chaque texture par les moindres carrés en utilisant l'algorithme génétique décrit précédemment. Nous avons choisi une population initiale importante en objets (comportant 20000 réalisations du vecteur de paramètres du modèle) afin de garantir la convergence vers le minimum global du critère employé.

Nous donnons comme illustration le résultat de modélisation des textures de la figure FIG. 2.19 (cf. FIG.2.20). Nous constatons d'une part, que le modèle choisi est adapté et d'autre part, que l'erreur d'estimation commise est relativement faible.

Nous avons récapitulé les paramètres du modèle de chacune des textures de la figure FIG. 2.19 dans les tableaux TAB. 2.7 et TAB. 2.8. On peut constater que l'erreur de modélisation commise est dans tous les cas faible et que les paramètres du modèle pour des textures proches sont relativement stables.

Les valeurs du caractère déterministe Δ_T pour chaque texture T sont cohérentes et correspondent bien à la perception visuelle. La valeur la plus faible est celle de la texture bruit (gaussien) et la plus grande, celle correspondant à la somme de sinusoïdes et du damier.

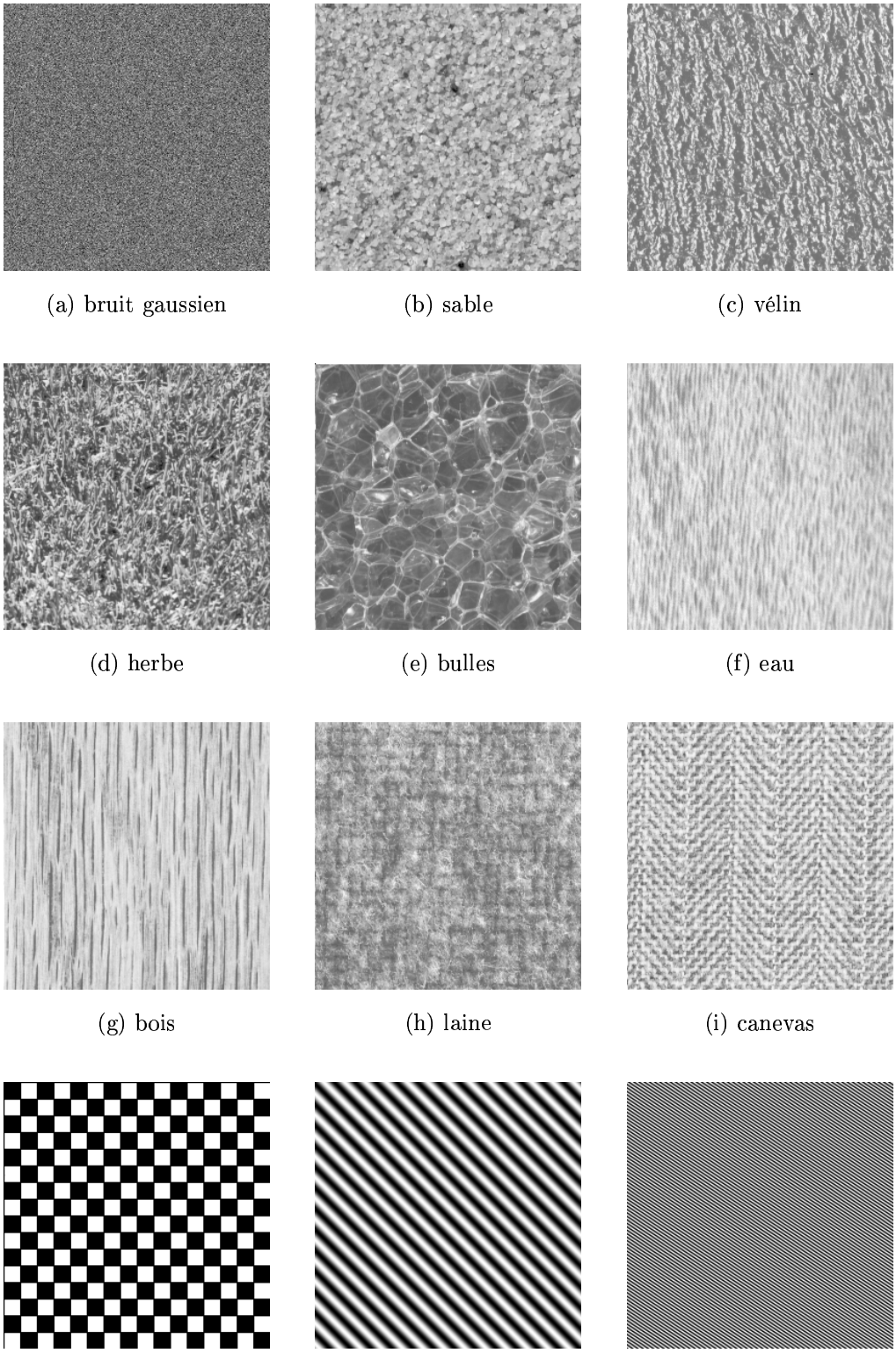
En ce qui concerne la finesse d'une texture, les valeurs obtenues pour les différentes textures tests sont également cohérentes. La valeur de la dimension du motif M_T la plus faible correspond au bruit gaussien et la plus élevée au damier.

Nous avons modélisé plusieurs textures à différentes orientations comme le montrent les figures FIG. 2.21 et FIG. 2.22. Les résultats de modélisation montrent l'invariance par rotation de la fonction F.

b) Analyse des paramètres du modèle et des attributs classiques

Nous avons essayé d'analyser la complémentarité de ces 8 nouveaux paramètres (α,γ,β,$T = 1/f$,ϕ,δ,Δ_T,M_T) avec les descripteurs de texture classiques retenus précédemment. Les résultats de cette étude sont consignés dans le tableau TAB. 2.9. Tout d'abord, on note une faible corrélation des paramètres du modèle. En effet, la corrélation moyenne de ces paramètres est égale à 36%.

Par ailleurs, les paramètres du modèle sont faiblement corrélés avec les descripteurs de texture retenus précédemment montrant ainsi leur complémentarité.



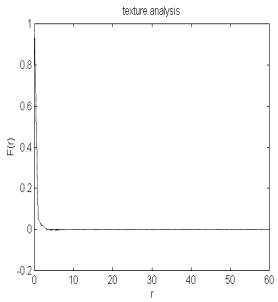
(a) bruit gaussien (b) sable (c) vélin

(d) herbe (e) bulles (f) eau

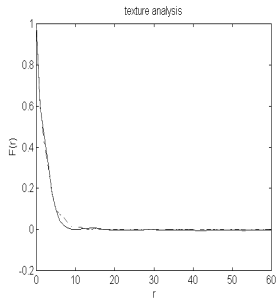
(g) bois (h) laine (i) canevas

(j) damier (k) sinusoïde (l) 2 sinusoïdes

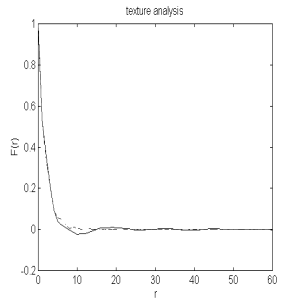
FIG. 2.19 – *Textures de Brodatz tests*



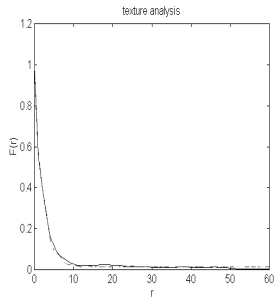
(a) bruit gaussien



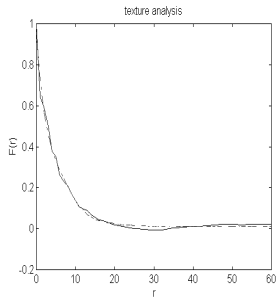
(b) sable



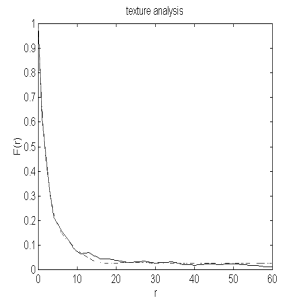
(c) vélin



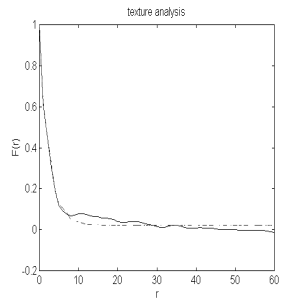
(d) herbe



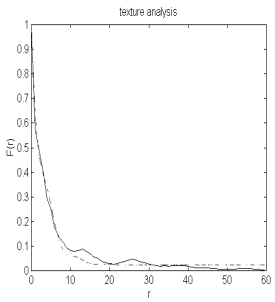
(e) bulles



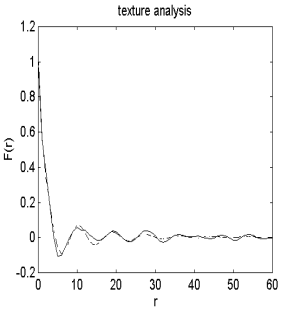
(f) eau



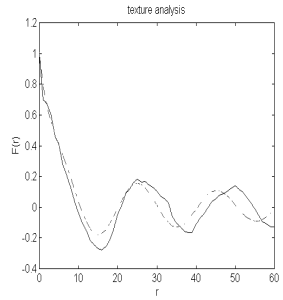
(g) bois



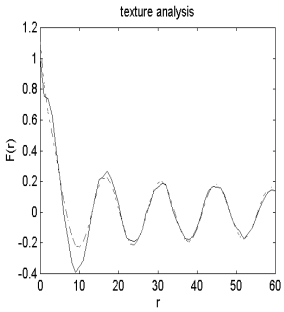
(h) laine



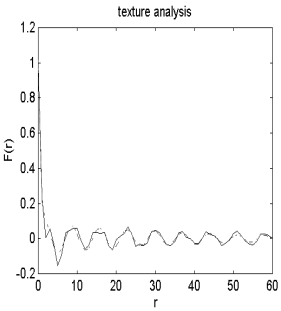
(i) canevas



(j) damier

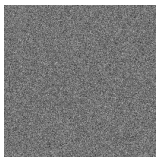
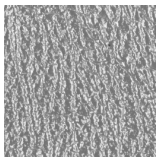
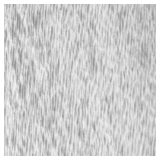
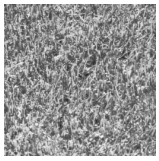
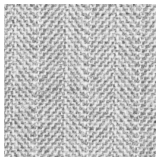
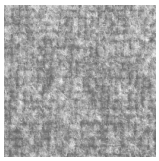


(k) sinusoïde

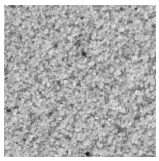
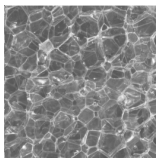
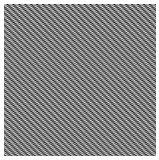
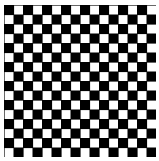


(l) 2 sinusoïdes

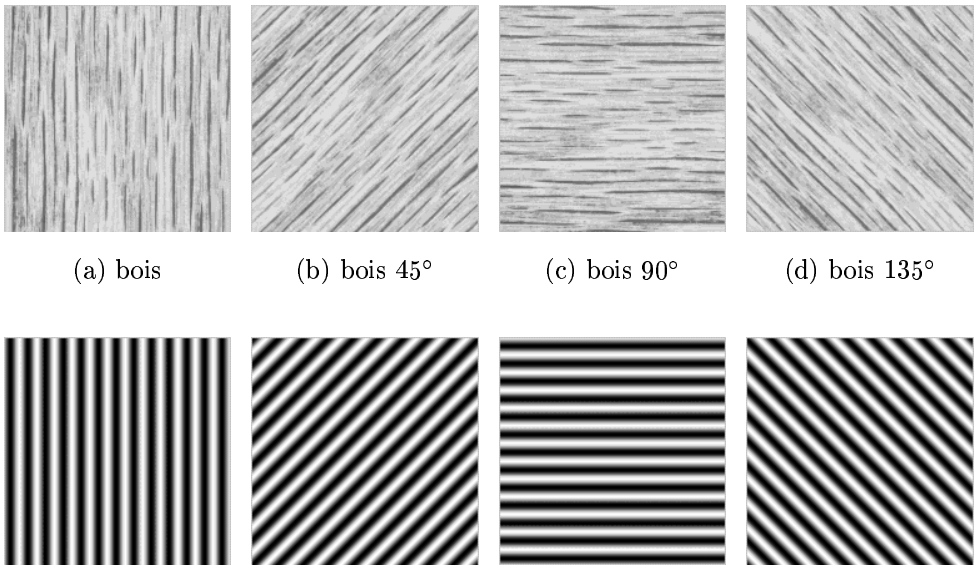
FIG. 2.20 – *Résultat de modélisation des Textures tests*

texture	α	γ	β	$T = 1/f$	ϕ	δ	ϵ	Δ_T	M_T
	3.21	0.02	0.25	7.17	4.70	0.00	0.00	0.00	3.00
	0.55	0.05	0.25	3.54	1.71	0.00	0.00	0.09	7.00
	0.42	0.07	0.25	3.89	1.91	0.02	0.03	0.10	9.00
	0.33	0.15	0.21	15.00	1.87	0.03	0.00	0.26	14.00
	0.47	0.07	0.25	3.66	1.68	0.01	0.00	0.11	9.00
	0.67	0.10	0.08	0.02	7.09	0.00	0.03	0.17	5.00
	0.32	0.16	0.25	7.23	1.87	0.02	0.03	0.24	14.00

TAB. 2.7 – *Paramètres du modèle pour les différentes textures tests*

texture	α	γ	β	$T = 1/f$	ϕ	δ	ϵ	Δ_T	M_T
	0.43	0.06	0.25	4.22	2.00	0.00	0.00	0.09	9.00
	0.22	0.15	0.25	13.19	1.91	0.01	0.02	0.25	16.00
	2.06	0.10	0.03	7.05	4.87	0.00	0.03	1	4.00
	0.46	0.26	0.00	14.08	5.12	0.00	0.17	1	16.00
	0.34	0.24	0.02	0.06	1.54	0.00	0.36	1	26.00

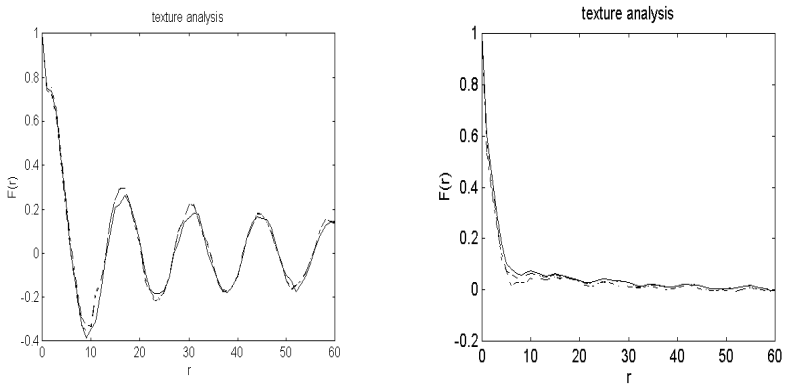
TAB. 2.8 – *Paramètres du modèle pour les différentes textures tests (suite)*



(a) bois (b) bois 45° (c) bois 90° (d) bois 135°

(e) sinus (f) sinus 45° (g) sinus 90° (h) sinus 135°

FIG. 2.21 – *2 textures pour 4 différentes orientations*



(a) sinus (b) bois

FIG. 2.22 – *Résultats de modélisation des textures à différentes orientations de la figure FIG. 2.21*

	α	γ	β	$T = 1/f$	ϕ	δ	Δ_T	M_T
cooc1	0.25	0.35	0.31	0.13	0.20	0.24	0.52	0.42
cooc2	0.13	0.08	0.22	0.12	0.15	0.11	0.29	0.16
cooc3	0.33	0.13	0.31	0.05	0.01	0.09	0.11	0.13
cooc4	0.02	0.03	0.10	0.14	0.14	0.10	0.27	0.13
cooc5	0.42	0.45	0.32	0.10	0.13	0.18	0.53	0.51
cooc6	0.19	0.17	0.16	0.20	0.01	0.09	0.02	0.09
cooc7	0.01	0.04	0.10	0.13	0.15	0.11	0.28	0.14
cooc8	0.16	0.32	0.22	0.13	0.25	0.21	0.50	0.34
cooc9	0.27	0.36	0.23	0.11	0.16	0.20	0.48	0.39
cooc10	0.38	0.42	0.35	0.12	0.13	0.20	0.54	0.54
cooc11	0.37	0.44	0.27	0.12	0.17	0.20	0.56	0.47
cooc12	0.67	0.57	0.45	0.09	0.06	0.16	0.50	0.67
cooc13	0.29	0.08	0.13	0.11	0.23	0.00	0.02	0.20
cooc14	0.44	0.47	0.35	0.12	0.11	0.21	0.53	0.52
cooc15	0.25	0.37	0.32	0.13	0.22	0.23	0.50	0.42
ldp1	0.47	0.45	0.30	0.10	0.08	0.20	0.52	0.48
ldp2	0.24	0.34	0.40	0.10	0.19	0.38	0.60	0.46
ldp3	0.00	0.09	0.19	0.03	0.06	0.03	0.11	0.04
ldp4	0.48	0.46	0.31	0.11	0.07	0.18	0.47	0.47
ldp5	0.44	0.47	0.35	0.12	0.11	0.21	0.53	0.52
hist1	0.04	0.12	0.05	0.04	0.07	0.16	0.07	0.04
hist2	0.12	0.17	0.05	0.05	0.10	0.12	0.06	0.04
ext1	0.44	0.28	0.29	0.03	0.03	0.01	0.36	0.40
ext2	0.44	0.36	0.28	0.01	0.00	0.02	0.42	0.45
ext3	0.58	0.44	0.31	0.02	0.03	0.17	0.43	0.52
ext4	0.40	0.36	0.30	0.01	0.05	0.02	0.43	0.42
curv1	0.08	0.24	0.23	0.13	0.01	0.11	0.16	0.16
curv2	0.09	0.25	0.22	0.13	0.01	0.11	0.17	0.17
curv3	0.10	0.26	0.21	0.13	0.02	0.08	0.19	0.20
curv4	0.08	0.25	0.22	0.11	0.00	0.04	0.20	0.17
α								
γ	0.52							
β	0.23	0.14						
$T = 1/f$	0.02	0.53	0.00					
ϕ	0.46	0.17	0.30	0.30				
δ	0.50	0.48	0.38	0.32	0.22			
Δ_T	0.26	0.50	0.53	0.03	0.39	0.31		
M_T	0.67	0.64	0.43	0.25	0.13	0.84	0.52	

TAB. 2.9 – *Corrélation des différents paramètres du modèle de textures*

La classification de tous les attributs de texture par la méthode CHAVL (cf. FIG. 2.23) montre d'une part la faible redondance des nouveaux attributs définis puisqu'ils ne sont agrégés qu'à des niveaux relativement grands. Ce résultat met en évidence d'autre part, la complémentarité des attributs classiques et les nouveaux attributs définis. En effet, les paramètres du modèle ont été regroupés dans un même sous-arbre (même si l'attribut *cooc*13 est dans ce sous-arbre).

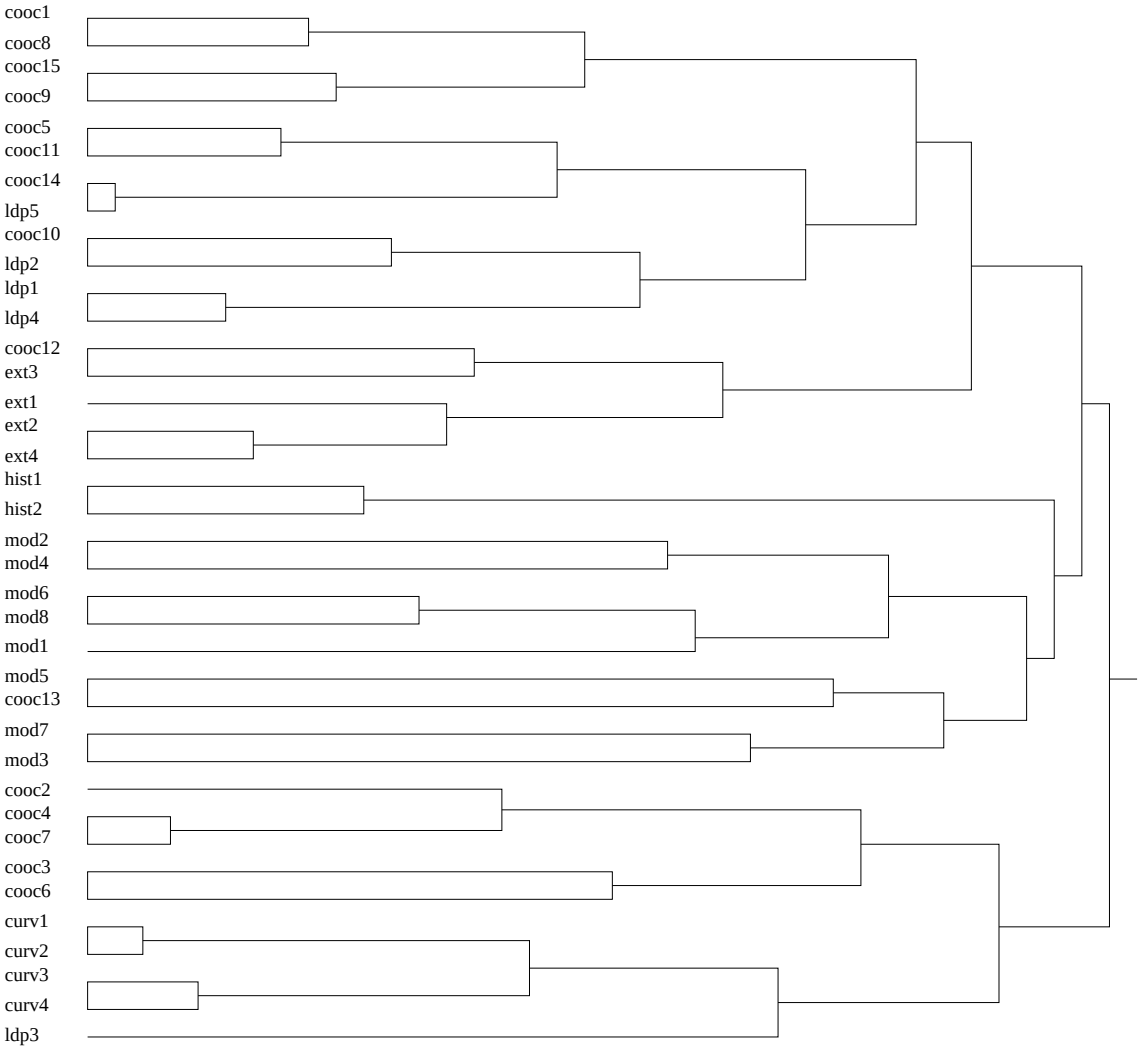
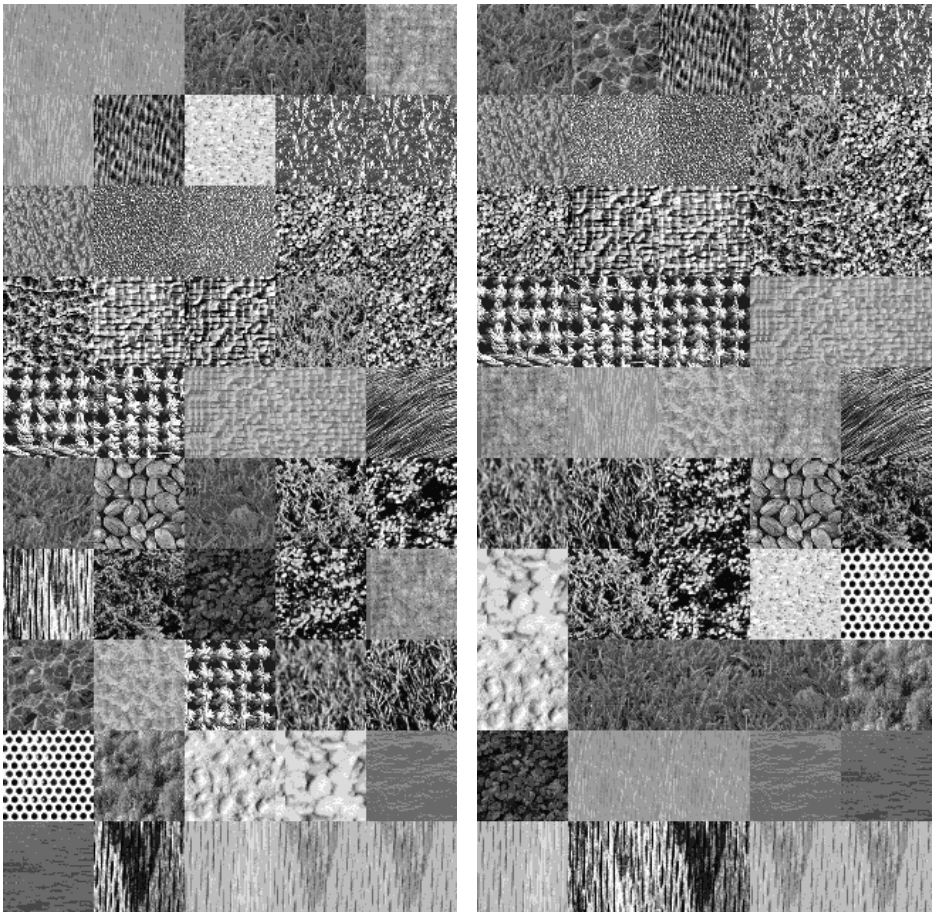


FIG. 2.23 – *Classification de tous les attributs de texture*

c) Classement des textres à partir de l'ensemble des paramètres

La méthode de classement est la même que celle employée précédemment (cf. section 2.3.1.3). Le résultat de classement des textures de la mosaïque (cf. FIG.

2.24(a)) par les paramètres du modèle peut d'un prime abord paraître moins bon que celui obtenu à partir des descripteurs de texture précédents. Ceci vient du fait que le modèle ne tient pas compte du contraste des textures mais uniquement de la corrélation des pixels d'une texture. On peut remarquer en outre que quatre des cinq textures bois de la mosaïque ont été regroupées ce qui n'était pas le cas auparavant.



(a) paramètres du modèle

(b) paramètres du modèle et descripteurs de texture

FIG. 2.24 – *Résultat de classement des textures de la mosaïque à partir des descripteurs de texture et des paramètres du modèle proposé*

Nous avons réalisé une classification supervisée des textures de la mosaïque en utilisant le critère de minimisation de distance euclidienne et les paramètres du modèle. La matrice de confusion est donnée dans le tableau TAB. 2.10. Le taux de bonne classification est meilleur que celui obtenu par les attributs de texture

(92.86%).

Les résultats de classement obtenus en combinant les descripteurs de texture avec les paramètres du modèle sont satisfaisants (cf. FIG 2.24(b)). Ils tiennent compte à la fois des propriétés statistiques de la texture tels que le contraste, la moyenne,.. et des propriétés de régularité et de finesse de la texture. Les textures visuellement proches sont bien regroupées. Le modèle de textures proposé permet d'enrichir la description statistique d'une texture en prenant en compte des composantes structurelles de celle-ci telles qu'une caractérisation du motif la composant et l'interaction entre ces motifs.

Nous avons classé les textures de cette base en mode supervisé en combinant les descripteurs de texture et les paramètres du modèle. La matrice de confusion est donnée dans le tableau TAB. 2.11. Le taux de bonne classification est le meilleur obtenu des deux précédentes expériences (93.48%) montrant l'intérêt de combiner ces deux types d'informations.

Nous avons réalisé le même type de classification sur une autre base de textures contenant 15 échantillons de 20 textures différentes de l'album de Brodatz soit 300 textures (cf. FIG. 2.25) en utilisant les descripteurs de texture classiques décrits précédemment avant et après réduction. Le taux de classification est de 86.6% dans les deux cas, ceci montre l'intérêt de réduire l'espace de mesures.

Le taux de classification obtenu en utilisant les paramètres du modèle proposé est correct (88.6%). Pour tenir compte des niveaux de gris (ce qui fait défaut aux paramètres du modèle), nous avons utilisé les moments d'ordre 1 à 4. Les résultats montrent d'une part, l'efficacité de la classification en n'utilisant que les moments d'ordre 1 à 4 (92% de bonne classification) et, d'autre part, l'apport du modèle (98% de bonne classification). Le tableau TAB. 2.12 récapitule tous les résultats de classification obtenus. Ce tableau montre qu'en combinant tous ces attributs (attributs classique après réduction, paramètres du modèle, moments d'ordre 1 à 4), le taux de classification est de 99%.

2.4 Conclusions

L'analyse de textures est un problème important en traitement d'images et reste un problème complexe. Nous avons essayé tout d'abord de faire le point sur les techniques existantes de description de textures en utilisant des attributs. Cette étude a permis de mettre en évidence la corrélation d'attributs de même type et de différents types. Nous avons en outre souligné l'importance de combiner différents attributs. Afin de compléter cette analyse, nous avons ensuite analysé une texture à partir de la fonction d'autocorrélation 2D. Nous en avons déduit

	d002	d004	d009	d019	d023	d024	d028	d029	d032	d038	d053	d057	d066	d068	d070	d074	d077	d084	d093	d112
d002	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d004	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d009	0	0	50	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
d019	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d023	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d024	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d028	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
d029	0	0	0	0	0	50	0	50	0	0	0	0	0	0	0	0	0	0	0	0
d032	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
d038	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
d053	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
d057	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
d066	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
d068	0	0	14.29	0	0	0	0	14.29	0	0	14.29	0	0	57.14	0	0	0	0	0	0
d070	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
d074	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
d077	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
d084	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
d093	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
d112	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

TAB. 2.10 – *Matrice de confusion en utilisant les paramètres du modèle : taux de bonne classification* 92.86%

	d002	d004	d009	d019	d023	d024	d028	d029	d032	d038	d053	d057	d066	d068	d070	d074	d077	d084	d093	d112
d002	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d004	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d009	0	0	62.50	0	0	12.50	0	12.50	0	0	0	0	0	0	0	0	0	0	0	12.50
d019	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d023	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d024	0	0	0	0	0	75.00	0	25.00	0	0	0	0	0	0	0	0	0	0	0	0
d028	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
d029	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
d032	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
d038	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
d053	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
d057	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
d066	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
d068	0	0	0	28.57	0	0	0	14.29	0	0	0	0	0	57.14	0	0	0	0	0	0
d070	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
d074	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
d077	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
d084	0	0	0	0	0	0	0	25.00	0	0	0	0	0	0	0	0	0	75.00	0	0
d093	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
d112	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

TAB. 2.11 – *Matrice de confusion en utilisant les attributs de texture et les paramètres du modele : taux de bonne classification* 93.48%

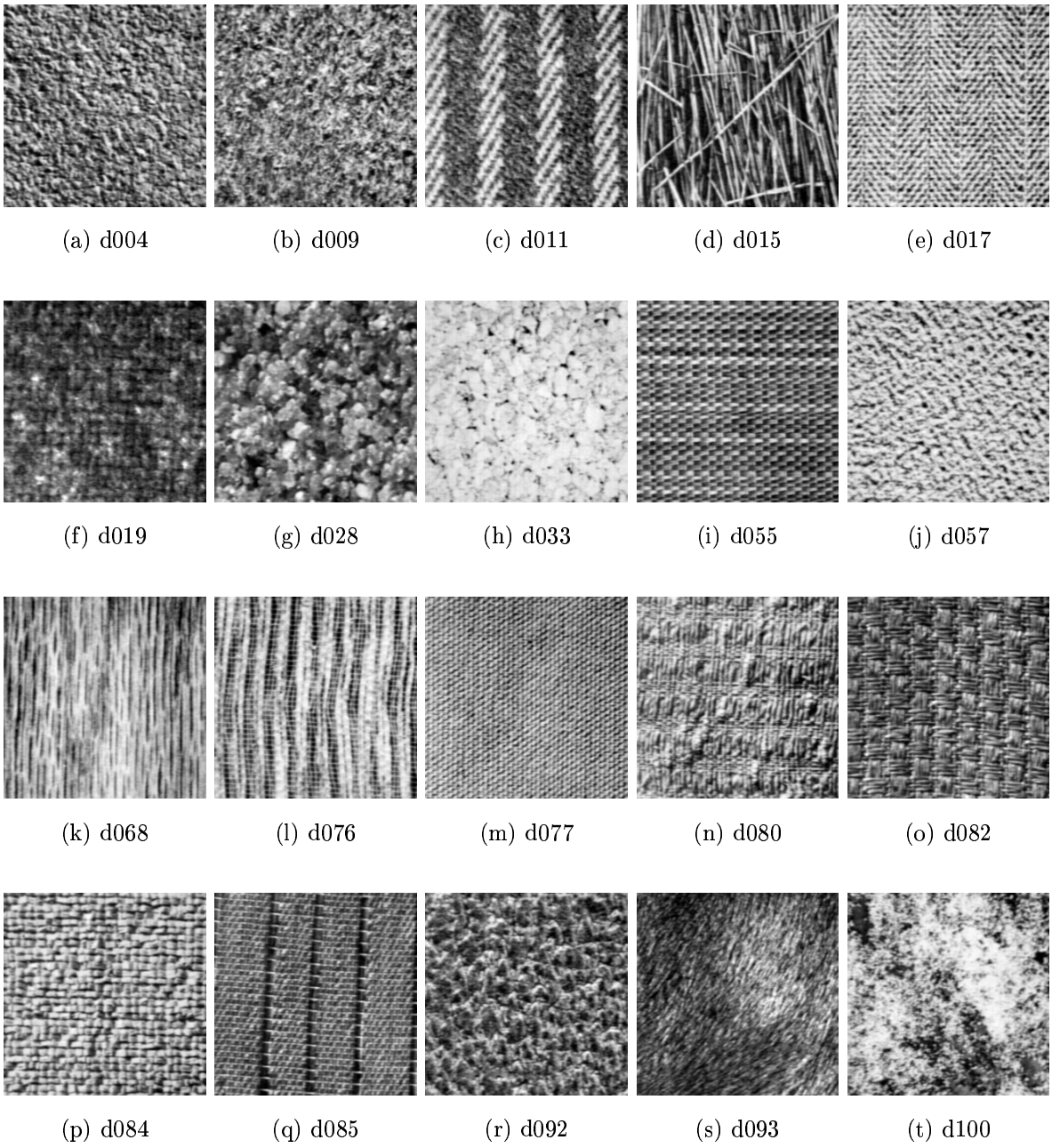


FIG. 2.25 – *20 Textures de Brodatz*

Attributs utilisés	Taux de classification
descripteurs de texture classiques (avant réduction)	86.6%
descripteurs de texture classiques (après réduction)	86.6%
moments d'ordre 1 à 2	72.3%
moments d'ordre 1 à 4	92.6%
paramètres du modèle	88.6%
paramètres du modèle + moments d'ordre 1 à 2	93.3%
paramètres du modèle + moments d'ordre 1 à 4	98%
paramètres du modèle + moments d'ordre 1 à 4 + descripteurs de texture classiques (après réduction)	99%

TAB. 2.12 – *Résultats de classification en utilisant différents types d'attributs*

une fonction 1D F traduisant la corrélation des pixels d'une texture dans toutes les directions. Cette propriété permet une analyse de texture invariante par rotation.

La fonction F a été modélisée en la considérant comme la fonction d'autocorrélation d'un signal 1D à partir de la décomposition de Wold. Les paramètres de ce modèle sont obtenus par moindres carrés à l'aide d'un algorithme génétique. Ainsi, il est possible de décrire une texture par un ensemble de 5 paramètres (α,β,γ,T,ϕ,δ). Les résultats expérimentaux sur un ensemble significatif d'images(62 textures) ont montré d'une part, que le modèle est bien adapté, et d'autre part, que l'erreur de modélisation est faible quel que soit le type de texture.

Nous avons défini un indice du caractère déterministe Δ_T d'une texture T à partir de 2 des paramètres de notre modèle (β,γ). Les résultats expérimentaux ont montré une bonne adéquation entre la valeur de cet indice et le caractère aléatoire ou déterministe de la texture.

Afin d'affiner cette caractérisation de texture, nous avons extrait un paramètre traduisant la finesse d'une texture. Cet attribut donne la taille minimale de la fenêtre d'analyse d'une texture. Cette information est primordiale en segmentation d'images puisqu'elle permettra d'adapter la taille du support d'analyse d'une texture.

Ces paramètres issus de la modélisation de la fonction d'autocorrélation 1D d'une texture sont complémentaires aux descripteurs de texture classique déterminés auparavant et aux moments d'ordre 1 à 4. Ils permettront une meilleure caractérisation d'une texture en vue soit de l'adaptation de la segmentation, soit de faciliter l'interprétation de régions texturées d'une image.

Le module d'analyse d'une image proposé permet de caractériser avec précision

les différentes régions la composant. Les zones texturées et uniformes de l'image sont tout d'abord extraites. Ensuite, la description des régions texturées est affinée afin de faciliter l'étape de segmentation.

Toutes ces informations extraites sur les différentes zones de l'image sont exploitées dans le module de segmentation du système proposé.

Chapitre 3

Segmentation adaptative

3.1 Introduction

Les informations extraites lors du module d'analyse de l'image permettent de choisir la méthode de segmentation adaptée au type de zone rencontré. Comme nous l'avons vu dans le chapitre 1, nous avons choisi d'utiliser l'approche de classification à partir d'attributs pour segmenter les différentes régions de l'image. Outre son efficacité, cette approche se distingue par sa généralité et son adaptation naturelle au contexte. En effet, les attributs utilisés pour classer les pixels d'une région, peuvent être choisis de telle sorte qu'ils permettent de différencier les pixels appartenant au type de région considéré. Par ailleurs, certains attributs permettent de choisir de manière adaptative la taille de la fenêtre d'analyse de la région dans la phase de segmentation.

Avant de détailler quels types d'attributs et quelle taille d'analyse ont été utilisés pour segmenter les zones uniformes et texturées de l'image à traiter, nous présentons la méthode de classification développée que nous avons utilisée.

3.2 Méthode de classification automatique

Les méthodes de classification automatique usuelles ne nécessitent pas d'apprentissage comme nous l'avons vu dans le chapitre 1, mais plusieurs paramètres doivent être définis comme le nombre de classes, le nombre de plus proches voisins ou les noyaux.

Les méthodes de classification présentées dans le chapitre 1 ne remettent pas en cause un résultat intermédiaire (partition courante) ou un choix arbitraire (noyaux). Ceci peut provoquer une mauvaise classification et une instabilité de la méthode. Pour pallier ces inconvénients, nous proposons une méthode de classification non

supervisée caractérisée par les propriétés suivantes :

- évaluation et possibilité de remise en cause d'un résultat intermédiaire,
- correction d'un résultat courant en exploitant les résultats antérieurs,
- évaluation de la solution sans connaissance de vérité de terrain.

La méthode de classification que nous proposons est dérivée de la méthode LBG (nuées dynamiques). Le point crucial des méthodes du type nuées dynamiques est l'initialisation. En effet, le choix des noyaux des classes influe beaucoup sur la convergence et la qualité du résultat. L'approche développée ne nécessite pas la connaissance du nombre de classes mais seulement d'un majorant.

On suppose disposer d'un ensemble de vecteurs d'attributs $\mathcal{A} = \{\mathbf{a}^1,..,\mathbf{a}_n\}$ et d'un ensemble d'individus $\mathcal{T} = \{T_1,..,T_n\}$ à classer. Pour déterminer la meilleure partition de $\mathcal{T}$, quatre étapes essentielles sont nécessaires (cf. FIG. 3.1) :

1. choix de la classe à subdiviser : à l'état initial ($k = 1$), la classe à subdiviser est l'ensemble $\mathcal{T}$. À chaque itération ($k > 1$), on essaie de partitionner la classe qui maximise une fonction de dispersion,
2. choix des noyaux de division d'une classe : une fois la classe à partitionner connue, on détermine les points de départ des nuées dynamiques en utilisant la méthode LBG,
3. modification de la partition à k classes en $k + 1$ classes par la méthode des nuées dynamiques,
4. évaluation de la partition de l'ensemble $\mathcal{T}$ suivant plusieurs critères. Si ces critères sont satisfaits (la partition de l'ensemble $\mathcal{T}$ à $k + 1$ classes est valide), alors l'étape 1 est relancée pour essayer de créér $k + 2$ classes. Dans le cas contraire, le choix des noyaux est remis en cause et l'étape 2 est de nouveau relancée en intégrant les échecs précédents. Si aucun noyau n'engendre une partition valide, le choix de la classe à subdiviser est remis en cause et l'étape 1 est relancée. Si aucune classe ne génère une nouvelle partition valide alors le nombre de classes et la partition obtenus sont optimaux.

Nous détaillons à présent chacune des étapes de ce processus.

3.2.1 Choix de la classe à subdiviser

Nous supposons avoir déterminé k classes $\{C_1,..,C_k\}$ et nous voulons essayer de diviser l'une d'entre elles pour en obtenir $k + 1$ au total. Intuitivement, la classe la plus étendue est la meilleure candidate à être partitionnée. On calcule donc une mesure de dispersion de chaque classe C_i, $i = 1,k$ de la façon suivante :

$$dispersion(C_i) = \frac{1}{n_i} \sum_{j=1}^{ni} d(B_i,C_i^j)$$

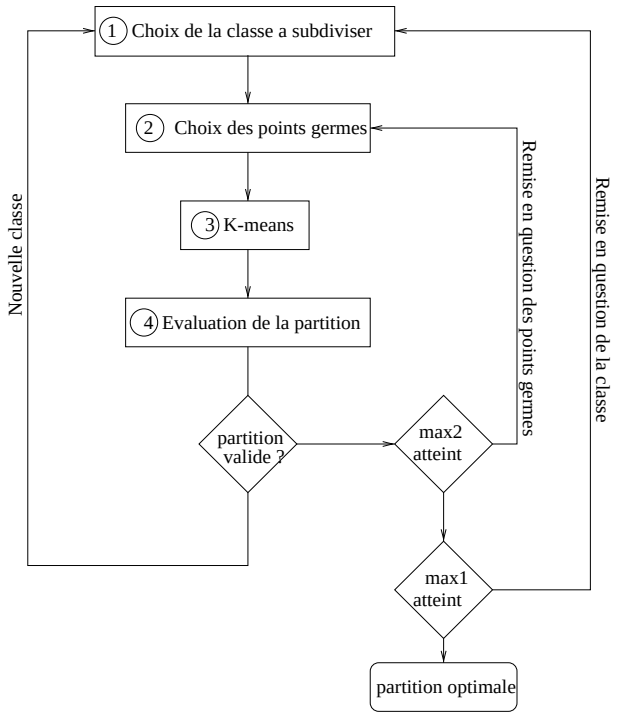


FIG. 3.1 – *Schéma des différentes étapes de la méthode de classification proposée*

où n_i est le nombre de vecteurs dans la classe C_i de barycentre B_i, $d(.,.)$ est la distance euclidienne et C_i^j est le j ème élément de la classe C_i.

On ordonne chacune des classes $\{C_1,..,C_k\}$ par ordre décroissant de leur mesure de dispersion. On essaie dans un premier temps de diviser la partition ayant la mesure de dispersion la plus élevée puis la suivante.

3.2.2 Choix des points germes

Une fois choisie la classe à partitionner C_k, nous devons déterminer les noyaux de la nouvelle classe. Cette classe sera divisée en deux. On se donne un nombre d'essais maximum de tentatives de subdivision de C_k noté max_1. Le point le plus représentatif de la classe C_k étant son barycentre B_k, on le conserve comme premier noyau G_1. Le choix du second est effectué de la façon suivante :

$$\begin{cases} G_2(i) = arg\ max_{j=1,n_k} d(B_k,C_k^j) & si\ i = 1 \\ G_2(i) = arg\ max_{j=1,n_k} (\sum_{l=1}^{i-1} d(C_k^j,G_2(l))) & si\ i < max_1 \end{cases}$$

On prend au premier essai ($i = 1$) le point le plus éloigné du barycentre. Pour les autres essais, on prend le plus éloigné des essais précédents, ceci pour éviter une redondance de calcul.

3.2.3 Nuées dynamiques (partitionnement)

Les noyaux étant définis, on applique un processus de type nuées dynamiques à l'ensemble des vecteurs de départ jusqu'à stabilité. Un nombre d'itérations maximal est fixé afin de décider si les noyaux initiaux sont corrects.

Soit $\{C_1,..,C_k\}$ les classes créées aux itérations précédentes de barycentres respectifs $\{B_1,..,B_k\}$, on suppose que la classe C_i a été choisie pour être divisée. Les points germes G_1 et G_2 ont été définis précédemment. On se fixe un nombre maximum d'itérations max_2 permettant d'évaluer la stabilité de la partition ainsi créée. L'algorithme de convergence vers une partition stable est le suivant :

1. Soient $\{\tilde{B}_1,..,\tilde{B}_{k+1}\}$ les barycentres courants de la partition de $\mathcal{T}$ à $k+1$ classes. A l'initialisation on a $\tilde{B}_j = B_j$, $j = 1..k$ et $\tilde{B}_{k+1} = G_2$,
2. Affectation de l'ensemble des individus de $\mathcal{T} = \{T_1,..,T_n\}$: un individu T_i est affecté à la classe C_j si

$$d(\mathbf{a}_i,B_j) = \min_{l=1..k+1} d(\mathbf{a}_i,B_l)$$

3. Calcul des barycentres $\{\tilde{B}_1,..,\tilde{B}_{k+1}\}$ des classes obtenues,
4. Recommencer en 2 tant qu'il n'y a pas stabilité des barycentres ou tant que le nombre d'itérations maximum max_2 n'est pas atteint.

Un exemple illustrant ce procédé est donné dans la figure FIG. 3.2. La convergence vers la solution attendue est ici atteinte en trois itérations.

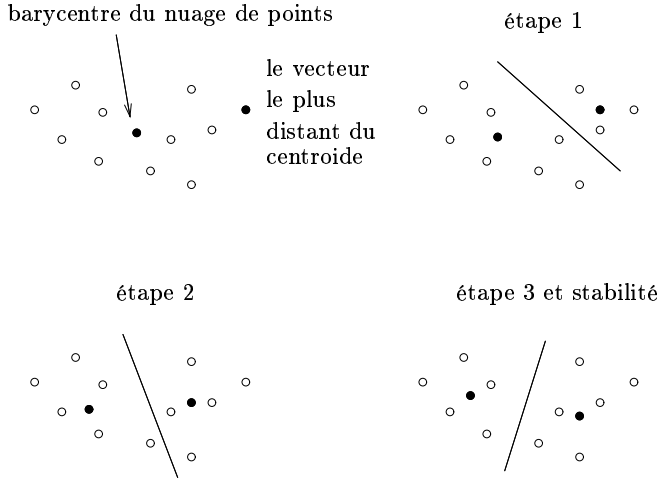


FIG. 3.2 – *Exemple de l'application de l'algorithme dans le cas de deux classes*

3.2.4 Évaluation d'une partition

Cette étape permet de valider ou de remettre en cause la partition ainsi créée en examinant plusieurs critères. Le premier concerne la stabilité de la partition. Si, au bout de max_2 itérations, il n'y a pas convergence, alors la partition est non valide car non stable.

Le second consiste à vérifier la cohérence d'une partition stable. Il s'agit d'évaluer la pertinence de la partition des données en $k+1$ classes en considérant son apport éventuel par rapport à celle en k classes. En effet, on veut ici détecter les partitions erronées (exemple de la figure FIG. 3.3). Il peut s'agir soit d'une classe partitionnée à tort ou d'une classe mal partitionnée.

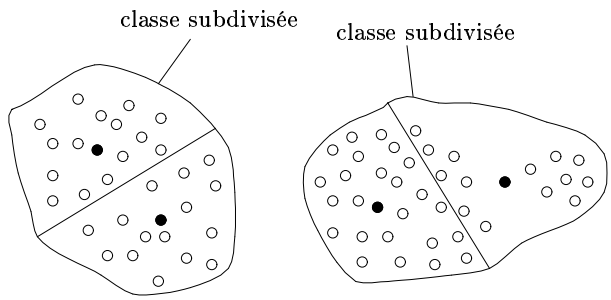


FIG. 3.3 – *Partitions erronées d'une classe*

La partition d'un ensemble en $k+1$ classes n'est justifiée que si la disparité intraclasse globale moyenne engendrée est inférieure à celle obtenue en k classes. La disparité intraclasse globale moyenne notée $\tilde{D}^{(k)}(\mathcal{T})$ d'une partition en k classes est définie comme suit :

$$\tilde{D}^{(k)}(\mathcal{T}) = \frac{1}{k.card(\mathcal{T})} \sum_{i=1}^{k} D(C_i).card(C_i)$$

où $D(C_i)$ est la disparité intraclasse de la classe C_i s'écrivant sous la forme :

$$D(C_i) = \frac{1}{card(C_i)} \sum_{T_j \in C_i} d(\mathbf{a}_j, B_j)$$

La partition en $k+1$ classes est validée si $\tilde{D}^{(k)}(A) - \tilde{D}^{(k+1)}(A) > \rho.\tilde{D}^{(k)}(A)$. La valeur de ρ est un pourcentage très faible garantissant l'arrêt de l'algorithme de subdivision des classes ($\rho = 0.01$ dans nos expérimentations).

L'évaluation de la partition est réalisée en considérant donc la stabilité et la cohérence des classes formées.

Deux cas peuvent se produire :

- une partition valide est obtenue, l'algorithme se poursuit en essayant de partitionner la classe la plus étendue parmi celles créées,
- dans le cas contraire, le choix du point germe est dans un premier temps remis en cause. Si, au bout de max_1 itérations, il n'y a pas convergence des partitions ou si les partitions résultantes ne sont pas valides, alors on remet en cause le choix de la partition à subdiviser et on en choisit une autre. Si enfin aucune partition ne peut être subdivisée, alors le nombre de classes et la partition obtenus sont optimaux.

3.2.5 Résultats expérimentaux

Nous présentons quelques résultats expérimentaux sur un ensemble d'attributs de dimension 2 afin que les résultats de classification soient faciles à analyser visuellement. Pour chaque type de test, l'ensemble de vecteurs a été engendré artificiellement de la façon suivante :

- on se fixe tout d'abord un nombre de classes $NCLASS$ ainsi que le nombre d'individus dans chaque classe N_i $i = 1,..,NCLASS$,
- on se donne un ensemble de points germes des classes $\{g_1,..,g_K\}$ où $g_i = (g_i^1,g_i^2)$,
- la connaissance des points germes de chaque classe permet de définir la distance de dispersion maximale autorisée d'un attribut pour une classe. Soit dm_i^j cette distance pour la classe i et pour l'attribut j, elle est définie de la façon suivante :

$$dm_i^j = |g_1^j - g_k^j|/2$$

  où $k = arg\ min(|g_i^j,g_k^j|)$.
- pour chaque classe $i = 1,..,NCLASS$, faire
  - pour chaque élément de la classe $j = 1,..,N_i$, faire
    - tirage aléatoire à partir d'une loi uniforme d'un nombre x dans [0 1],
    - le j ème vecteur d'attributs de la classe i est défini comme suit :

$$\mathbf{a}_j^i = ((2x-1)dm_i^1,(2x-1)dm_i^2)$$

    - refaire
  - refaire

Le premier exemple est un nuage de points concentré (cf. FIG 3.4(a)) ne comportant a priori qu'une classe. Notre méthode de classification ne détecte qu'une seule classe (cf. FIG 3.4(b)).

Le second exemple est un nuage de points allongé constitué a priori d'une seule classe (cf. FIG 3.4(c)). Notre algorithme détecte ici 10 classes (cf. FIG 3.4(d)). Le majorant du nombre de classes était 10. Ceci montre les limites de l'algorithme

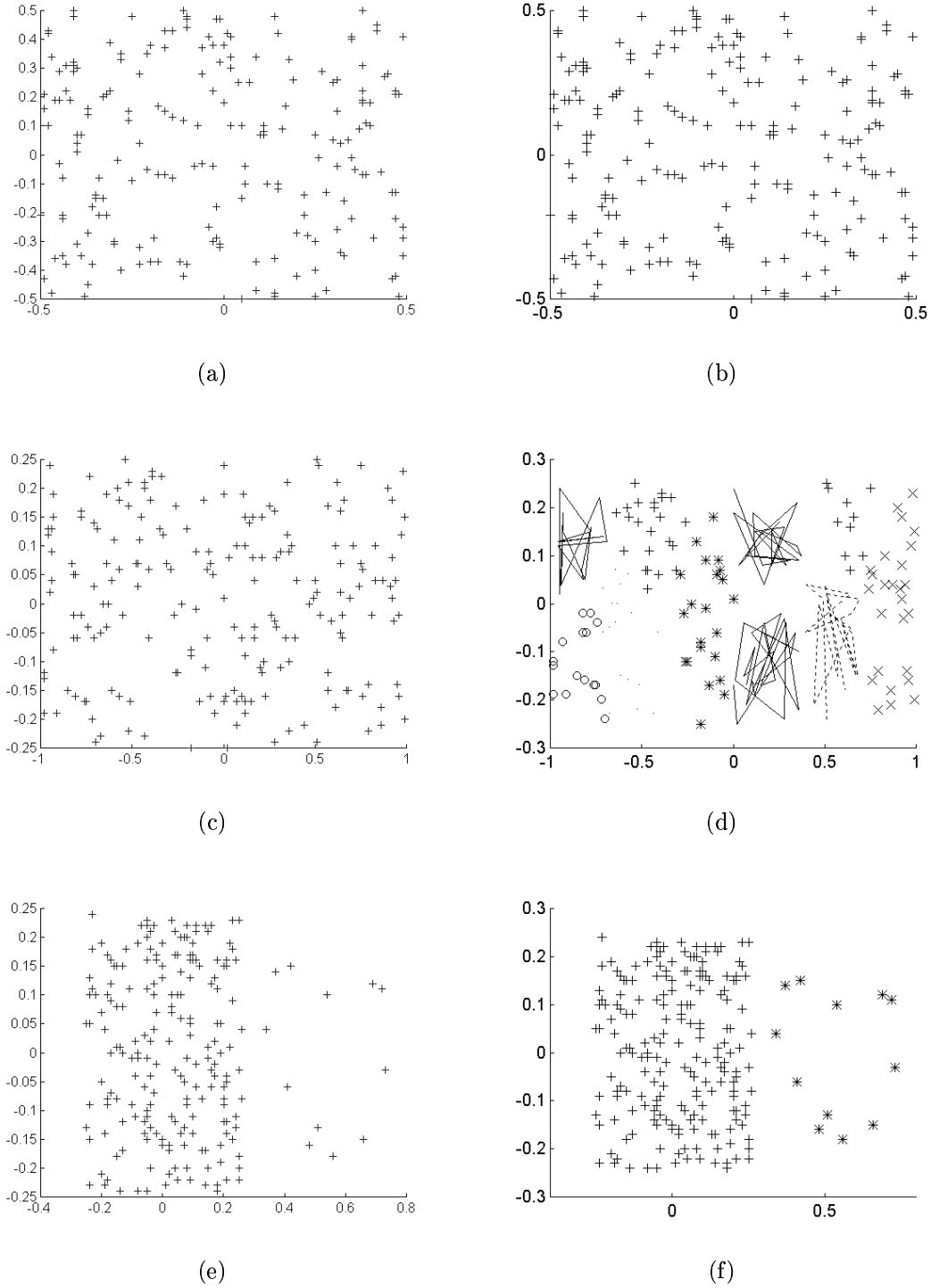


FIG. 3.4 – *Résultat de classification d'un nuage de points composé d'une classe et 2 classes*

dans ce cas extrême.

Nous avons testé l'algorithme en prenant un nuage de points constitué de deux classes, l'une contenant beaucoup plus d'éléments (cf FIG. 3.4(e)). Le nombre de classes détecté est deux et chaque élément est correctement classé.

Nous présentons enfin deux résultats de classification sur des nuages de points contenant six classes (cf. FIG. 3.5). Nous avons comparé les résultats de la méthode poposée avec une méthode classique de nuées dynamiques floues [Chi94],[Bez81]. Nous avons réalisé la classification des deux nuages de points précédents en donnant le bon nombre de classes. Les résultats obtenus montrent l'efficacité de notre approche en ce qui concerne la discrimination des six classes.

Nous allons intégrer cette méthode de classification automatique dans notre système pour segmenter les images.

3.3 Segmentation

Nous allons détailler la ou les méthodes de segmentation employées pour traiter chaque type de régions de l'image.

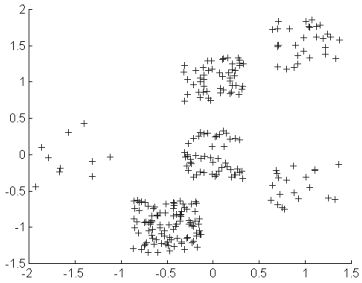
3.3.1 Segmentation des zones uniformes

Les méthodes possibles pour segmenter ce type de régions sont nombreuses : parmi celles-ci on peut citer les méthodes de croissance de régions [CJA96], [Buf89], les méthodes de détections de contours [MLT95], [Can86] et les méthodes de multi-seuillage avec fusion de régions [KCC95]. Dans la mesure où nous voulons utiliser une méthode de segmentation générique, nous avons utilisé une méthode de classification à partir d'attributs. Cette méthode est efficace et s'adapte très facilement en choisissant les attributs pertinents pour segmenter ce type de régions.

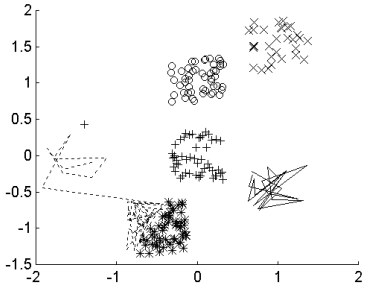
Chaque fenêtre W de taille $NW \times NW$ pixels (NW impair) centrée sur le pixel à classer est caractérisée par un vecteur d'attributs (cf. FIG. 3.6).

Dans le cas où le pixel $s \in Bord = I - \check{I}$ ($\check{I} = [\frac{NW-1}{2} + 1, NLIG - \frac{NW-1}{2}] \times [\frac{NW-1}{2} + 1, NCOL - \frac{NW-1}{2}]$), le vecteur d'attributs de ce pixel est pris égal à celui du pixel $s' \in \check{I}$ le plus proche.

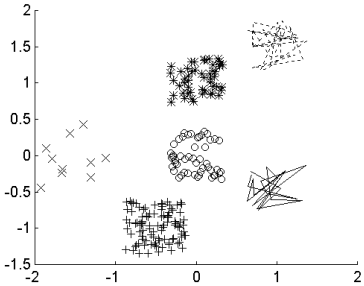
Nous avons choisi de caractériser un pixel d'une zone uniforme par la moyenne et la variance des niveaux de gris pour une fenêtre d'analyse de taille 3×3 pixels. Cette information est suffisante pour segmenter une région uniforme tout en préservant d'éventuels détails.



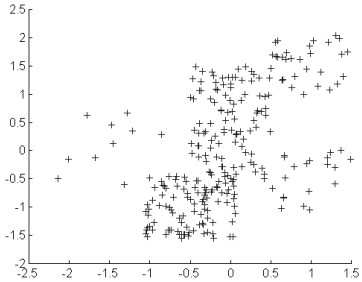
(a)



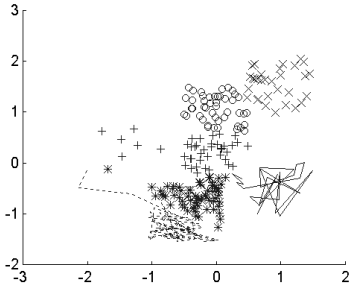
(b) Fuzzy C-means



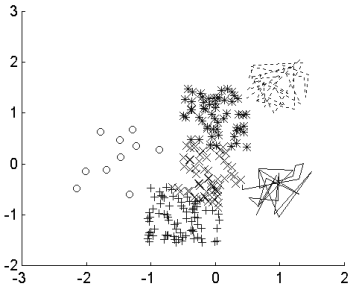
(c) méthode proposée



(d)



(e) Fuzzy C-means



(f) méthode proposée

FIG. 3.5 – *Résultat de classification d'un nuage de points composé de 6 classes par l'algorithme proposé et l'algorithme de nuées dynamiques floues*

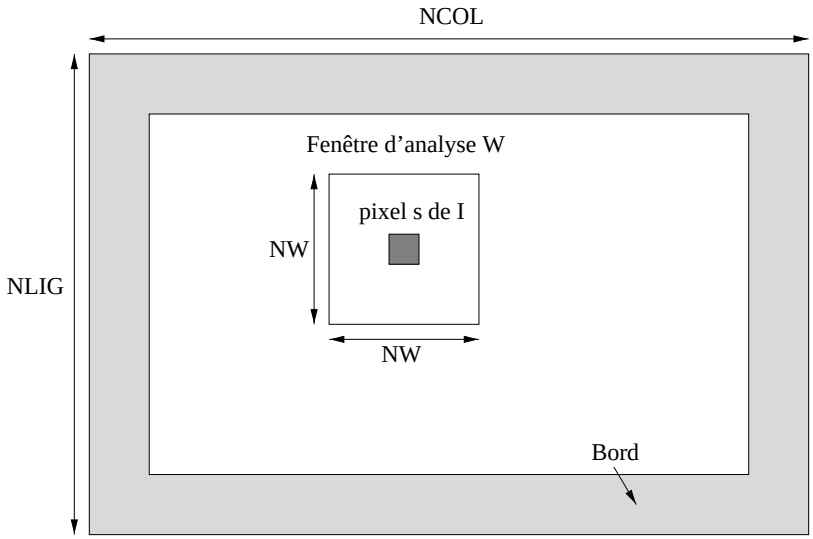


FIG. 3.6 – *Description d'un pixel de l'image sur une fenêtre d'analyse*

3.3.2 Segmentation des zones texturées

Le schéma de segmentation est identique à celui concernant les zones uniformes de l'image. Pour segmenter correctement les régions texturées, nous adoptons la taille de la fenêtre d'analyse en fonction du contenu informationnel local de la région analysée, puis introduisons les attributs de texture décrits dans le module 1 du système:

- *la taille de la fenêtre d'analyse*: elle est déterminée à partir de l'analyse des régions texturées décrite précédemment (cf. section 2.3.2.6). Chaque pixel d'une zone texturée est caractérisé par le modèle décrit précédemment. Ensuite, une classification des pixels d'une zone texturée est réalisée à partir du vecteur de paramètres du modèle. Ceci permet de mettre en évidence un nombre de régions texturées de granularité et de caractère déterministe connus. Chaque région localisée est segmentée en utilisant la taille de la fenêtre appropriée, c'est à dire 3×3 dans le cas d'une région uniforme et $M_T \times M_T$ dans le cas texturé,
- *les attributs*: ils correspondent aux descripteurs de texture précédents après réduction, les paramètres du modèle que nous avons définis et les moments d'ordre 1 à 4.

Une fois chaque type de zone segmenté, plusieurs types de post-traitements sont nécessaires afin d'obtenir le résultat final :

- traitement des points isolés : il peut arriver que la segmentation produise des régions de très petite taille ne contenant que quelques pixels (moins de 10 pixels). Afin de régler ce problème, nous appliquons un processus de relaxation sur l'image segmentée,

– labelisation des régions de l'image segmentée : afin de rendre plus interprétable l'image segmentée, chaque classe est représentée par un label correspondant à la moyenne des niveaux de gris des pixels de l'image dans cette classe.

3.4 Résultats expérimentaux

Pour tester la validité du système proposé, nous avons tout d'abord effectué des tests sur les images utilisées précédemment (cf. FIG 2.4).

Les figures FIG. 3.7 et 3.8 montrent les résultats de segmentation obtenus par le système développé. Les images composées majoritairement de régions uniformes sont correctement segmentées. Néanmoins, dans le cas d'images plutôt texturées, on remarque une certaine sur-segmentation. Il faut noter que la méthode employée pour segmenter chaque type de zone est très simple mais présente l'avantage d'être assez rapide en temps de calcul. Afin d'améliorer le résultat, il serait possible d'utiliser d'autres approches plus élaborées sur les zones texturées afin de localiser plus finement les frontières des régions.

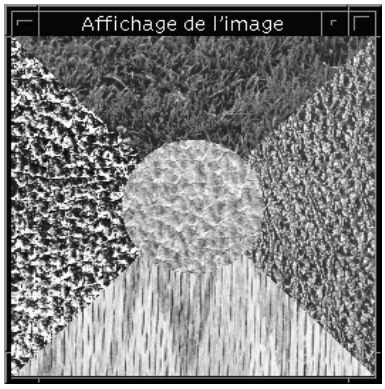
Ensuite, nous avons utilisé des images du CASI dont la résolution est plus fine. Cette dernière application entre dans le cadre des travaux du G.S.T.B. (Groupement d'intérêt Scientifique de Télédétection en Bretagne). Afin de mener à bien les actions de recherche liés à l'environnement, le G.S.T.B. s'est doté d'un capteur multispectral aéroporté : le C.A.S.I. (Compact Airborne Spectrographic imager). Le système permet d'acquérir des images dans la gamme de longueurs d'onde allant de 430 à 870 nm avec un maximum de 19 bandes spectrales distinctes.

Nous donnons le résultat de segmentation de 6 images obtenues par le C.A.S.I. (cf. FIG. 3.9 et FIG. 3.10) sur une bande spectrale. Les trois premières images ont une taille de 512×512 pixels et les trois dernières 256×256 pixels.

Les résultats de segmentation sont visuellement corrects. Pour valider ces résultats, une vérité de terrain est nécessaire.

3.5 Conclusions

Les résultats de segmentation obtenus par ce module sont corrects. Les méthodes de segmentation utilisées pour traiter chaque type de zone pourraient être plus complexes afin d'améliorer les résultats de segmentation. La classifcation à partir d'attributs présente ici l'avantage d'être générale, facilement paramétrable afin de s'adapter au contexte et est assez économique en temps de calcul.



(a) image synthétique : SYNT1

(b) résultat de segmentation

(c) image synthétique : SYNT2

(d) résultat de segmentation

(e) image scène d'extérieur : EXT1

(f) résultat de segmentation

FIG. 3.7 – *Segmentation adaptative des images SYNT1, SYNT2 et EXT1*

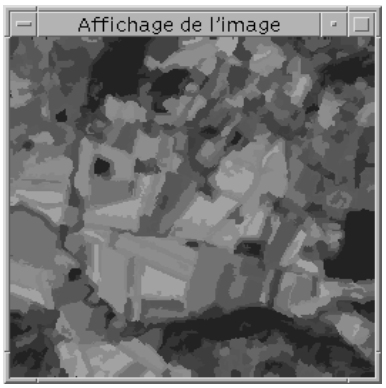
(a) image scène d'extérieur : EXT2

(b) résultat de segmentation

(c) image Télédétection : TELE1

(d) résultat de segmentation

(e) image Télédétection : TELE2



(f) résultat de segmentation

FIG. 3.8 – *Segmentation adaptative des images EXT2, TELE1 et TELE2*

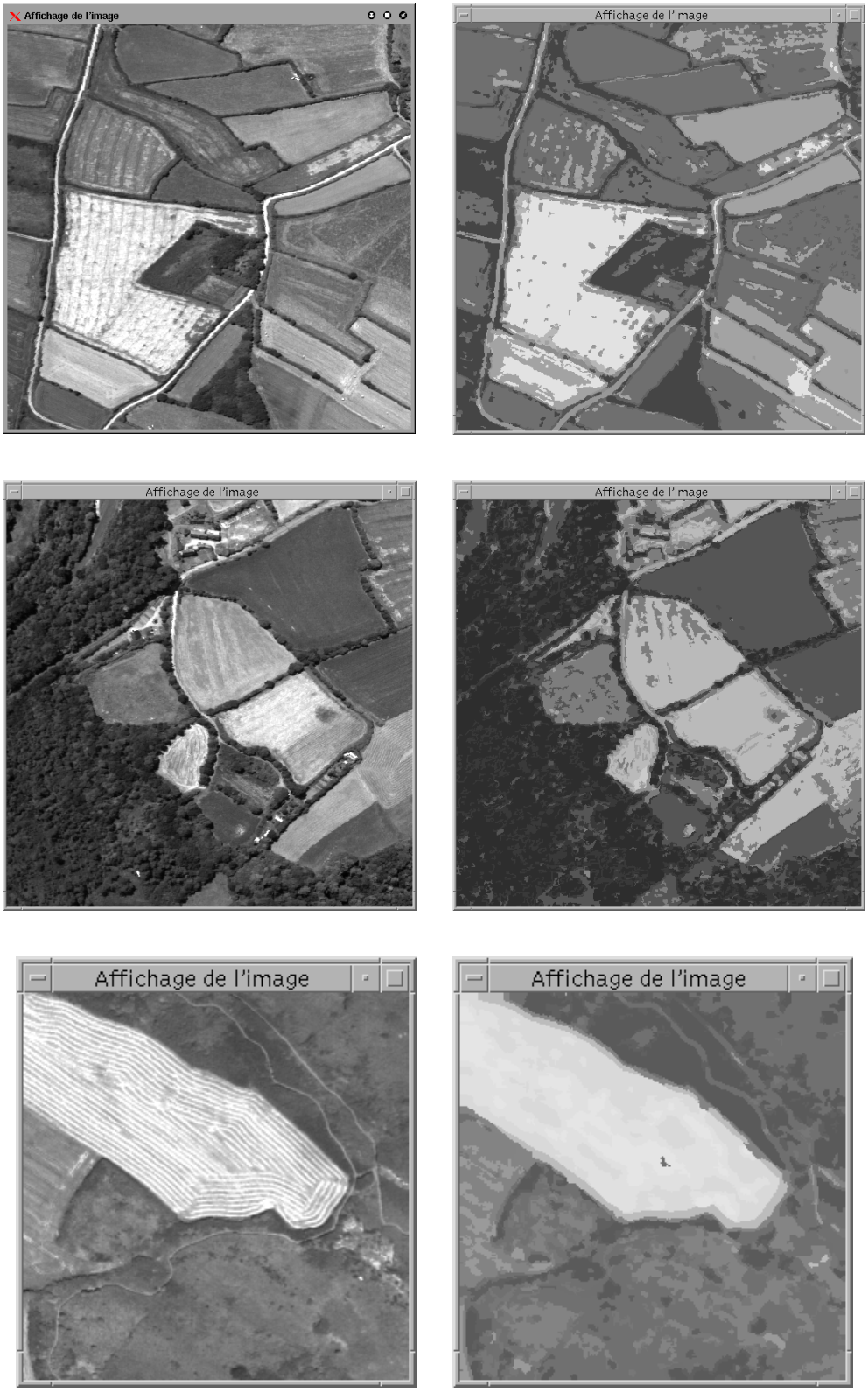


FIG. 3.9 – *Résultats de segmentation de quelques images acquises par le C.A.S.I*

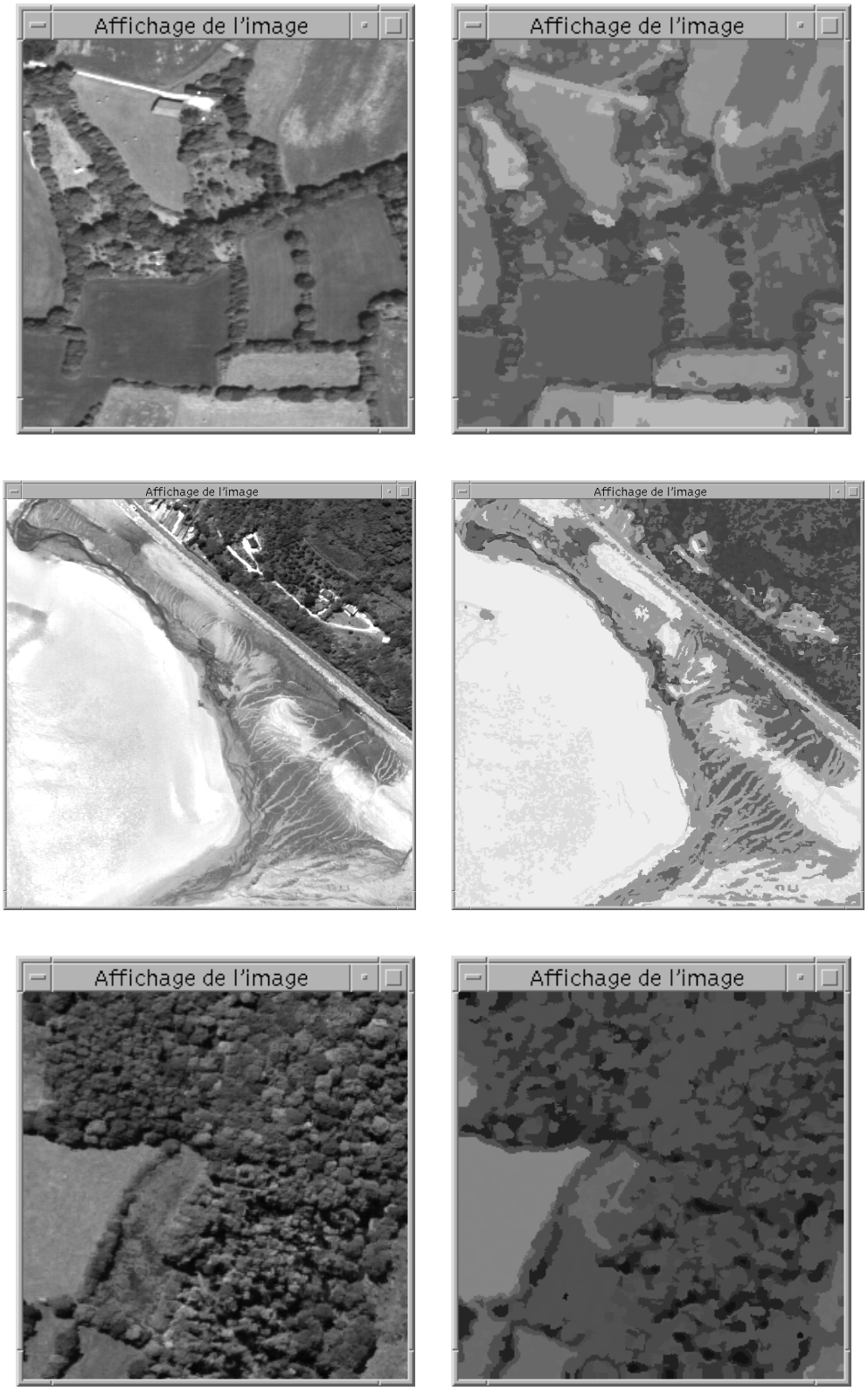


FIG. 3.10 – *Résultats de segmentation de quelques images acquises par le C.A.S.I.*

L'étape de segmentation ne traitant pas des images multi-composantes, nous segmentons donc les images bande par bande, puis dans le module que nous allons détailler dans le chapitre suivant, les résultats de chaque bande peuvent être fusionnés.

Chapitre 4

Fusion de résultats de segmentation

4.1 Introduction

La fusion a pour but de combiner différents résultats de segmentation d'une même image ou de plusieurs bandes dans le cas d'images multi-composantes.

Les techniques existantes de fusion tentent d'exploiter soit la redondance de différents résultats de segmentation [fB95], [BHDP95], soit leur complémentarité [Laf96], [RB95] notamment en adoptant une approche multirésolution. Ces approches ont le principal inconvénient de ne pas intègrer la qualité de chaque résultat de segmentation mais uniquement leur corrélation.

Huet et al [HP98] propose néanmoins une méthode de fusion supervisée de résultats de segmentation basée sur l'interprétation. La fusion est réalisée en considérant la mesure de confiance dans l'appartenance d'un objet à une classe de la base d'apprentissage. Cette approche nécessite la connaissance des objets ou classes à identifier dans la scène (par exemple batiments, arbres,..), ce qui la rend inadaptée dans le cadre d'une chaîne automatique de segmentation d'images.

Nous proposons ici une méthode de fusion génétique en intégrant une information sur la qualité de chaque résultat de segmentation. Nous détaillons dans un premier temps le principe général d'un algorithme génétique. Nous formalisons dans un second temps la méthode de fusion génétique que nous proposons. Nous détaillons le critère d'évaluation d'un résultat de segmentation proposé. Nous présentons par la suite quelques résultats expérimentaux et nous conluons.

4.2 Les algorithmes génétiques

Les algorithmes génétiques déterminent les solutions d'une fonction en simulant l'évolution d'une population jusqu'à la survie des individus les plus adaptés

[Wal96]. Les survivants sont des individus obtenus par croisement, mutation et sélection d'individus de la génération précédente. Un algorithme génétique est défini en considérant cinq données essentielles :

1. *la représentation de la solution ou génotype* : chaque individu d'une population est décrit par son génotype. Le génotype est un ensemble de caractéristiques de l'individu tels que sa taille, son poids,
2. *la population initiale* : elle comprend un ensemble d'individus, chacun décrit par son génotype. Cette population peut être soit générée aléatoirement soit prédéfinie,
3. *la fonction d'aptitude* : elle permet de mesurer l'aptitude d'un individu en fonction de son génotype,
4. *les opérations sur les génotypes* : elles définissent les modifications des génotypes afin de faire évoluer une population au cours des générations. Il existe trois types d'opérations :
   - mutation d'un individu : les gènes d'un individu sont légèrement modifiés afin de mieux s'adapter à l'environnement,
   - sélection d'un individu : les individus inadaptés à l'environnement ne survivent pas à la génération suivante,
   - croisement ou reproduction de deux individus : deux individus peuvent se reproduire. Les gènes de l'individu résultant est une combinaison des deux parents,
5. *critère d'arrêt de l'évolution de la population* : il permet de mettre fin à l'évolution d'une population en considérant l'aptitude du meilleur individu ou du nombre de générations écoulées.

Une fois définies ces cinq données, l'éxecution d'un algorithme génétique s'effectue en 5 étapes :

1. définition de la population initiale et calcul de l'aptitude de chaque individu,
2. sélection et mutation des individus de la population courante,
3. reproduction des individus restant de la population courante,
4. évaluation des individus de la population,
5. revenir en 2 si le critère d'arrêt n'est pas satisfait.

Les algorithmes génétiques permettent de résoudre des problèmes de recherche opérationnelle ne pouvant pas être résolus par des techniques classiques de manière rapide.

Pour fusionner les résultats de segmentation, nous avons utilisé cet algorithme génétique. Dans cet algorithme, nous avons introduit un critère d'évaluation que nous avons développé comme fonction d'aptitude. Ce critère quantifie de manière globale la qualité d'un résultat de segmentation sans connaissance *a priori* d'une

vérité terrain. Dans ce qui suit, nous allons détailler sa définition, puis préciser les différentes données utilisées dans l'algorithme génétique.

4.3 Critère d'évaluation d'un résultat de segmentation

Dans le cas général, aucune vérité terrain n'est disponible pour évaluer un résultat de segmentation. Si l'on se réfère à la littérature, il n'existe pas à notre connaissance de méthode générale d'évaluation non supervisée de la qualité d'une segmentation. Certaines méthodes existantes ne sont pas adaptées en particulier aux régions texturées et nécessitent la plupart du temps la vérité de terrain [Zha96b].

Nous proposons ici une méthode d'évaluation de la qualité d'un résultat de segmentation ne nécessitant aucune connaissance *a priori*, basée sur la définition de la segmentation par Haralick [Har79]. Le critère défini [RC98] sera exploité dans la méthode de fusion proposée afin de pondérer chaque résultat de segmentation. Ce critère peut être aussi utilisé afin d'évaluer un résultat de segmentation lorsque la vérité de terrain n'est pas disponible. Nous mettons en évidence la performance de cette technique au travers de quelques résultats expérimentaux.

4.3.1 Définition du critère

L'objectif de l'évaluation est, étant donnés une image et un ensemble de méthodes de segmentation, de déterminer la ou les méthodes donnant le meilleur résultat au sens d'un critère défini. Le principe utilisé, lorsque la vérité de terrain n'est pas connue, consiste à estimer l'homogénéité des régions de l'image segmentée. Ceci est réalisé, la plupart du temps, en considérant l'écart-type de celle-ci. Or, une région texturée peut avoir un écart-type élevé et ne contenir qu'une seule texture.

Dans l'élaboration d'un critère d'évaluation d'un résultat de segmentation, on doit tenir compte des deux propriétés suivantes [Har79] :

- Une région de l'image segmentée ne doit contenir qu'une seule primitive (une texture ou un niveau de gris constant), ceci pour garantir qu'il n'y ait pas sous-segmentation. Ainsi, une région est caractérisée par une stabilité des statistiques en son sein.
- Deux régions limitrophes doivent contenir deux primitives différentes pour garantir qu'il n'y ait pas sur-segmentation. Ceci correspond à une disparité des statistiques entre ces deux régions [Zha96b].

En tenant compte de ces deux points, on peut considérer que le meilleur résultat de segmentation I^* d'une image donnée I parmi n au sens d'un critère F que nous

précisons par la suite est tel que :

$$I^* = arg \max_{j=1..n} F\left(\underline{D}(I^j)\ ,\ \overline{D}(I^j)\right)$$

La disparité intraclasse globale $\underline{D}(I^j)$ quantifie l'homogénéité de chacune des régions de l'image I^j. De même, la disparité interclasse globale $\overline{D}(I^j)$ mesure la disparité globale de chacune des régions de l'image I^j.

Pour avoir une valeur quantitative d'un résultat de segmentation, nous avons choisi la fonction F suivante pour le calcul de notre critère :

$$F\left(\underline{D}(I),\overline{D}(I)\right) = \frac{\overline{D}(I) + 1 - \underline{D}(I)}{2}$$

Ce critère correspond à une maximisation de la vraisemblance d'une partition.

L'écriture de ce critère répond bien à nos attentes, à savoir qu'une sous-segmentation sera pénalisée au travers de la disparité intraclasse globale qui sera forte, et la sur-segmentation induira une disparité interclasse globale faible diminuant le critère. La valeur du critère est normalisée car les disparités intraclasse et interclasse le sont comme nous le verrons.

Le nombre de régions composant l'image segmentée I^j peut varier selon la méthode employée. On note m_j le nombre de régions R_i^j, $i = 1..m_j$ de l'image segmentée I^j, r_i^j le nombre de pixels de la région R_i^j et NT le nombre de pixels total de l'image.

La disparité intraclasse globale $\underline{D}(I^j)$ d'une image segmentée I^j traduit la stabilité des statistiques de chacune des régions de l'image. Elle est calculée à partir de la disparité intraclasse $\underline{D}(R_i^j)$ [1] des différentes régions R_i^j, $i = 1..m_j$ de l'image segmentée :

$$\underline{D}(I^j) = \frac{1}{m_j} \sum_{i=1}^{m_j} \frac{r_i^j}{NT}\ \underline{D}(R_i^j)$$

L'importance de la disparité intraclasse d'une région R_i^j dans le calcul de la disparité intraclasse globale est proportionnelle au nombre de pixels r_i^j de cette région. Il est en effet souhaitable qu'une région de petite taille ait une influence moindre dans le calcul de la disparité intraclasse globale. Une région contenant deux primitives différentes doit avoir une disparité intraclasse élevée en rapport avec la même région composée d'une seule primitive.

1. Cette notion de disparité intraclasse n'est pas nouvelle, elle a été notamment utilisée dans les méthodes d'agrégation de régions [CP95].

Le principe est identique pour le calcul de la disparité interclasse globale $\overline{D}(I^j)$ de l'image segmentée I^j qui mesure la disparité de chaque région avec ses voisines :

$$\overline{D}(I^j) = \frac{1}{m_j} \sum_{i=1}^{m_j} \frac{r_i^j}{NT} \overline{D}(R_i^j)$$

Celle-ci est calculée à partir de la disparité interclasse $\overline{D}(R_i^j)$ de chaque région R_i^j. Le calcul de chacun de ces deux critères sera précisé par la suite.

4.3.2 Disparité intraclasse

On essaie ici d'évaluer l'homogénéité d'une région, c'est-à-dire la variation des statistiques à l'intérieur de celle-ci. Dans le calcul de disparité intraclasse, nous prenons en compte la nature de la région segmentée, à savoir uniforme ou texturée. La première étape consiste donc à déterminer si la région considérée est uniforme au sens des niveaux de gris ou si elle est texturée.

Le critère d'appartenance à une région texturée utilisé est celui défini dans le chapitre 2 en utilisant le paramètre uniforme U.

4.3.2.1 Régions uniformes

Dans le cas uniforme, la disparité intraclasse d'une région est donnée par l'écart-type normalisé[2] de celle-ci :

$$\underline{D}(R_i) = \tilde{\sigma}(R_i) = \frac{2}{255}\sqrt{\frac{1}{r_i}\sum_{(k,l)\in R_i} I_{(k,l)}^2 - \frac{1}{r_i^2}(\sum_{(k,l)\in R_i} I_{(k,l)})^2)}$$

où $r_i = card\{(k,l) \in R_i\}$

Cette statistique d'ordre 2 sur la dispersion des niveaux de gris d'une région est suffisante pour caractériser la disparité intraclasse d'une région uniforme.

4.3.2.2 Régions texturées

Si la région est texturée, l'écart-type seul ne donne pas d'information suffisante sur l'homogénéité de celle-ci. Nous proposons alors un processus plus complexe :

- Pour chaque région R_i de l'image segmentée, Faire :
  - Analyse locale de la région par balayage d'une fenêtre d'exploration W de taille $NW \times NW$ (cf. FIG 4.1) avec chevauchement maximal. Soit

2. L'écart type maximal d'une population à valeurs dans {0,1,..,254,255} est égal à 255/2. Si N est le nombre total d'individus, il correspond au cas où $N/2$ individus ont la valeur 0 et $N/2$ individus ont la valeur 255. La moyenne dans ce cas vaut 255/2 et l'écart à la moyenne est aussi égal à 255/2

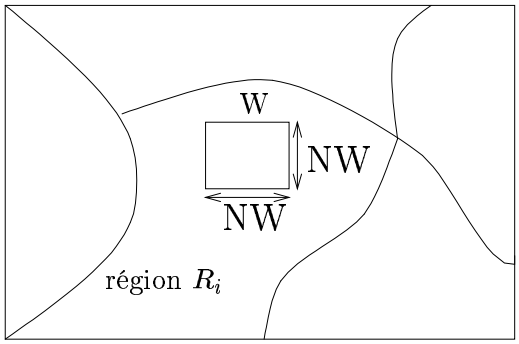


FIG. 4.1 – *Découpage en blocs de l'image à traiter*

$in_i = card(W \cap R_i)$, si $\frac{in_i}{NW^2} > 80\%$ alors on estime que l'information à l'intérieur de la fenêtre est significative (ce seuil permet de définir une fenêtre d'analyse adaptative dont la taille est suffisante),

- Calcul d'un vecteur d'attributs $\mathbf{a}$ sur cette fenêtre significative W. Ce vecteur est composé des attributs de texture retenus dans le chapitre 2 soit 27 paramètres.

- Refaire
- Pour chaque région R_i, faire
  - Soit $\mathcal{A} = \{\mathbf{a}^1,..,\mathbf{a}^{l(i)}\}$ l'ensemble de l vecteurs caractéristiques de la région R_i pour $l(i)$ fenêtres, on calcule le barycentre de la région R_i de la façon suivante :

$$B^i = \frac{1}{l(i)} \sum_{1 \leq p \leq l} \mathbf{a}^p$$

  - Détermination de la disparité intraclasse de la région R_i : On réalise une classification en 2 classes de cette région par un algorithme de K-means. Afin d'estimer la disparité intraclasse de la région R_i, on analyse ce résultat de classification. Soient c_1 et c_2 les deux barycentres des classes résultantes P_1 et P_2. On définit la mesure de dispersion m d'une classe d'éléments dans $\mathcal{T} = \{T_1,..,T_{card(T)}\}$ de barycentre B comme suit:

$$m = \frac{1}{card(\mathcal{T})} \sum_{i=1}^{card(\mathcal{T})} d(B,a_i)$$

Cette valeur quantifie l'écart moyen d'un élément de la classe à son barycentre. On pose m_1 et m_2 les mesures de dispersion de chacune des partitions P_1 et P_2.
On définit la disparité intraclasse de la façon suivante (cf. FIG. 4.2):

$$\underline{D}(R_i) = \begin{cases} 1 & si\ d(B_1,B_2) > m_1 + m_2 \\ 0 & sinon \end{cases}$$

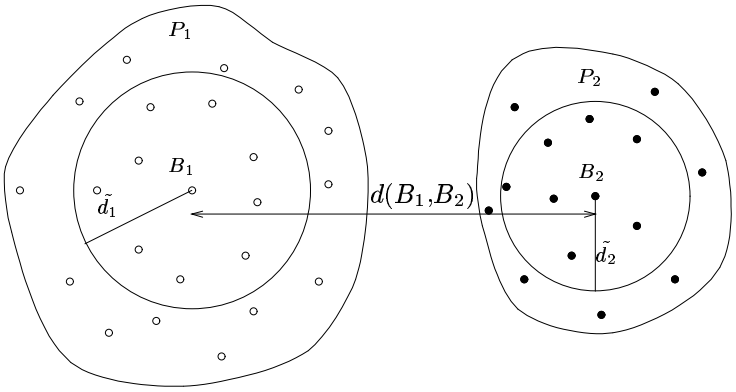


FIG. 4.2 – *Définition de la disparité intraclasse*

Si ces deux classes sont relativement séparées, alors la disparité intraclasse est pénalisée par la valeur maximale 1 puisqu'il existe une partition valide au sens de ce critère de la région R_i. Dans le cas contraire, la région R_i est considérée comme homogène et donc sa disparité intraclasse est nulle.

– Traitement des points isolés : Soit out_i le nombre de points isolés de la région R_i (n'appartenant pas à la composante connexe maximale de la région), la disparité intraclasse est mise à jour de la façon suivante :

$$\underline{D}(R_i) = \underline{D}(R_i) + \frac{out_i}{NT}$$

Ceci permet de prendre en compte dans le calcul de la disparité intraclasse d'une région, les pixels n'appartenant pas à la composante connexe maximale de celle-ci.

- Refaire

La disparité intraclasse d'une région est calculée à partir de la dispersion des attributs d'une région. Nous avons utilisé les 29 attributs de texture utilisés précédemment (cf. chapitre 2) pour caractériser chaque région de l'image segmentée.

Le traitement des points isolés permet de prendre en compte des régions de très petite taille n'ayant pas en général de réelle signification. On pénalise la disparité intraclasse de cette région par le pourcentage de points isolés de l'image. En effet, une région texturée de petite taille n'a pas vraiment de sens, il est donc légitime d'augmenter sa disparité intraclasse.

4.3.3 Disparité interclasse

Le procédé d'évaluation de la disparité interclasse d'une région est similaire à ce qui a été réalisé précédemment. En effet, au lieu d'estimer l'homogénéité d'une

région, on calcule sa disparité avec les régions voisines. La disparité interclasse s'écrit dans ce cas, comme la moyenne de la disparité d'une région avec ses voisines.

Il existe deux types de régions dans une image :

- région uniforme,
- région texturée caractérisée par un ensemble de vecteurs décrits précédemment.

Le problème se pose lors du calcul de la disparité interclasse, de quantifier la disparité entre régions de même classes et de types différents. Nous abordons dans un premier temps le cas de figure le plus simple.

4.3.3.1 Disparité entre régions de type différent

Il est clair que la disparité entre une région uniforme et une région texturée doit être maximale, elle est donc égale à 1 (cette mesure est normalisée).

4.3.3.2 Disparité entre régions de même type

Le calcul de la disparité entre deux régions de même type est plus délicat. Une région est décrite par la valeur moyenne des statistiques représentatives de la primitive. Ainsi, une région uniforme est décrite par sa moyenne et une région texturée par le vecteur d'attributs moyen, c'est à dire son barycentre. On estime alors la disparité entre deux primitives en considérant l'écart entre ces valeurs moyennes.

Tout d'abord, on calcule la disparité D entre deux régions uniformes R_i et R_j de la façon suivante :

$$D(R_i,R_j) = \frac{|E[R_i] - E[R_j]|}{NG}$$

où $E[R_i]$ est le niveau de gris moyen de la région R_i et NG est le nombre de niveaux de gris dans l'image. Cette valeur estime l'écart normalisé entre les niveaux de gris moyens de deux régions uniformes limitrophes. Cette information est suffisante dans la mesure où une région uniforme a, en général, une moyenne constante.

La disparité entre deux régions texturées R_i et R_j se calcule de la façon suivante :

$$D(R_i,R_j) = \frac{d(B_i,B_j)}{||B_i|| + ||B_j||}$$

où $d(.,.)$ est la distance euclidienne et B_i est le barycentre de la région R_i.

On note $\mathcal{V}_i = \{R_1,..,R_{q^{(i)}}\}$ l'ensemble des régions voisines de R_i. La disparité interclasse de cette région s'exprime sous la forme :

$$\overline{D}(R_i) = \frac{1}{q^{(i)}} \sum_{j=1}^{q^{(i)}} D(R_i,R_j)$$

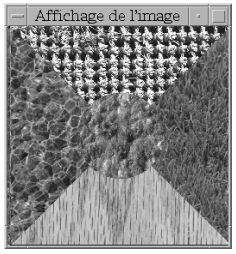
Suivant le type de primitive de chaque région, on adapte le calcul de la dissimilarité en fonction du contexte. La disparité interclasse d'une région exprime la dissimilarité moyenne des statistiques d'une région avec ses voisines.

4.3.4 Validation du critère

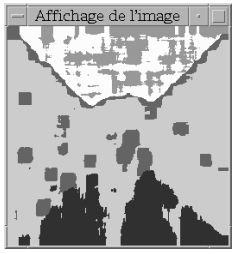
Nous avons réalisé deux types de tests expérimentaux pour valider le critère d'évaluation proposé. Le premier concerne la vérification de la cohérence du critère d'évaluation avec le taux de bonne classification de résultats de segmentation pour des images dont la vérité de terrain est connue. Le deuxième type de test est une comparaison de différents résultats de segmentation en appliquant le critère d'évaluation proposé sans connaissance de la vérité de terrain.

Nous avons comparé 10 résultats de segmentation de 5 images synthétiques dont les régions sont connues *a priori* (cf. FIG. 4.3). Ces 5 images comportent des types de régions relativement différentes (régions faiblement bruitées, fortement et faiblement texturées). Les résultats de segmentation ont été obtenus en utilisant une méthode de classification d'attributs par un algorithme Kmeans classique. La taille de la fenêtre d'analyse est de 15 pour chaque résultat. Les premiers résultats de segmentation (deuxième colonne dans la figure FIG. 4.3) ont été obtenus en utilisant la moyenne et la variance comme attributs. Les seconds résultats de segmentation (troisième colonne dans la figure FIG. 4.3) ont été obtenus en utilisant les attributs de cooccurrence et de longueur de plages. Même si certains résultats de segmentation sont de mauvaise qualité, nous avons souhaité les conserver afin d'apprécier le comportement du critère proposé.

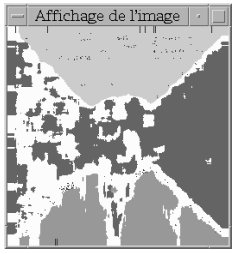
Le tableau TAB. 4.1 récapitule la valeur du critère pour chaque résultat de segmentation ainsi que le taux de bonne classification associé. Avant de commenter ces résultats, il faut préciser que deux valeurs du critère ne peuvent être comparées que pour la même image originale. En effet, ce critère tient compte des stabilités et des dissimilarités de stastitiques des différentes régions de l'image à segmenter. Si l'on analyse les variations du critère d'évaluation et du taux de bonne classification, on remarque qu'elles sont cohérentes les unes par rapport aux autres. En effet, si un résultat de segmentation est meilleur qu'un autre au sens du taux de bonne classification, il l'est aussi en considérant la valeur du critère d'évaluation. Certes, la variation du critère d'évaluation n'est pas proportionnelle à celle du taux de bonne classification. Le critère d'évaluation se comporte de façon cohérente par



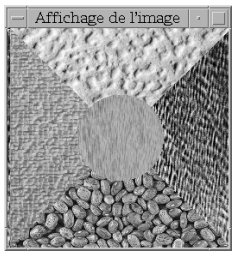
(a) image test1



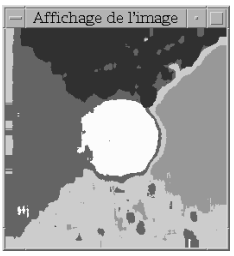
(b) $test1_1$



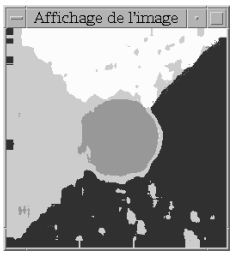
(c) $test1_2$



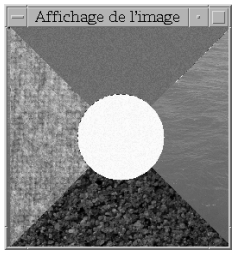
(d) image test2



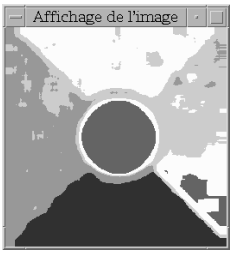
(e) $test2_1$



(f) $test2_2$

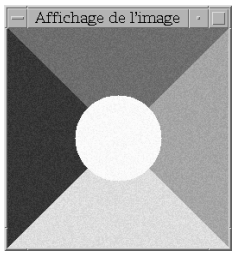


(g) image test3

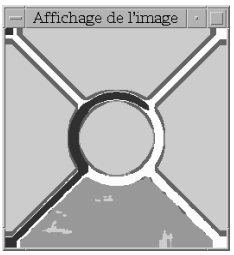


(h) $test3_1$

(i) $test3_2$

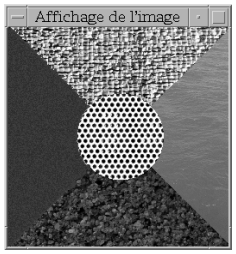


(j) image test4

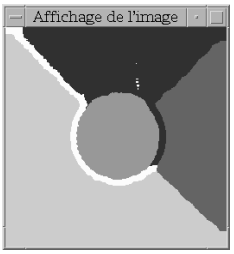


(k) $test4_1$

(l) $test4_2$



(m) image test5



(n) $test5_1$

(o) $test5_2$

rapport au taux de bonne classification. Nous allons donc l'appliquer afin d'évaluer des résultats de segmentation dont la vérité de terrain n'est pas connue.

Image	Disparité intraclasse	Disparité interclasse	Critère	Taux de classification
$test1_1$	0.26	0.92	0.88	41.6%
$test1_2$	0.04	0.80	0.92	58.2%
$test2_1$	0	0.61	0.87	86.2%
$test2_2$	0.28	0.63	0.78	73%
$test3_1$	0.31	0.81	0.83	78.8%
$test3_2$	0.68	0.8	0.7	54.6%
$test4_1$	0.52	0.67	0.71	20.8%
$test4_2$	0.37	0.59	0.73	44%
$test5_1$	0.58	0.95	0.68	75%
$test5_2$	0.16	0.73	0.85	93.8%

TAB. 4.1 – *Critère d'évaluation et taux de bonne classification des résultats de segmentation de la figure FIG. 3.17*

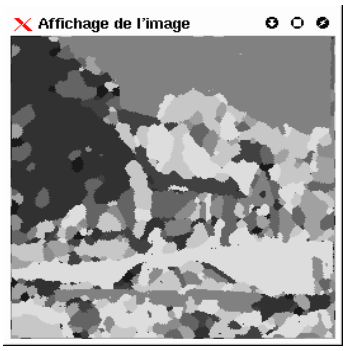
Nous avons évalué 3 résultats de segmentation de deux images CAR et GSTB01 (cf. FIG. 4.4, FIG. 4.5). La vérité de terrain n'est pas connue, il n'est donc pas possible de calculer le taux de bonne classification. Le premier résultat de segmentation CAR_1 de l'image CAR a été obtenu par une méthode bayésienne utilisant le SEM [CC96]. Le deuxième résultat CAR_2 a été obtenu en utilisant l'étape de segmentation adaptative du système décrit dans le chapitre 3. Enfin, nous avons utilisé une methode classification d'attributs (moyenne, variance) pour le résultat CAR_3. Visuellement, le résultat CAR_1 montre que l'image est sur-segmentée, ce qui se traduit par une valeur de la disparité intraclasse assez forte (cf. TAB. 4.2). Dans le résultat CAR_3, l'image est sous-segmentée (valeur de la disparité interclasse importante). Le résultat CAR_2 est un bon compromis car il donne la meilleure valeur du critère.

Image	Disparité intraclasse	Disparité interclasse	Critère
CAR_1	0.58	0.84	0.75
CAR_2	0.36	0.8	**0.81**
CAR_3	0.40	0.57	0.72

TAB. 4.2 – *Évaluation des résultats de segmentation de l'image "CAR"*

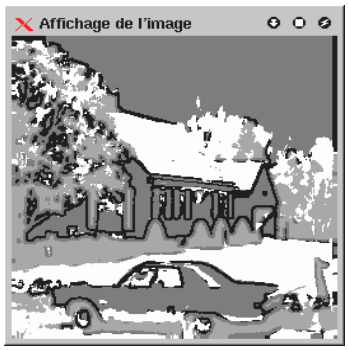
L'image GSTB01 a été segmentée en utilisant deux types de méthodes : une méthode de classification d'attributs (moyenne, variance) avec deux tailles de fenêtre

(a) image CAR



(b) image CAR_1

(c) image CAR_2



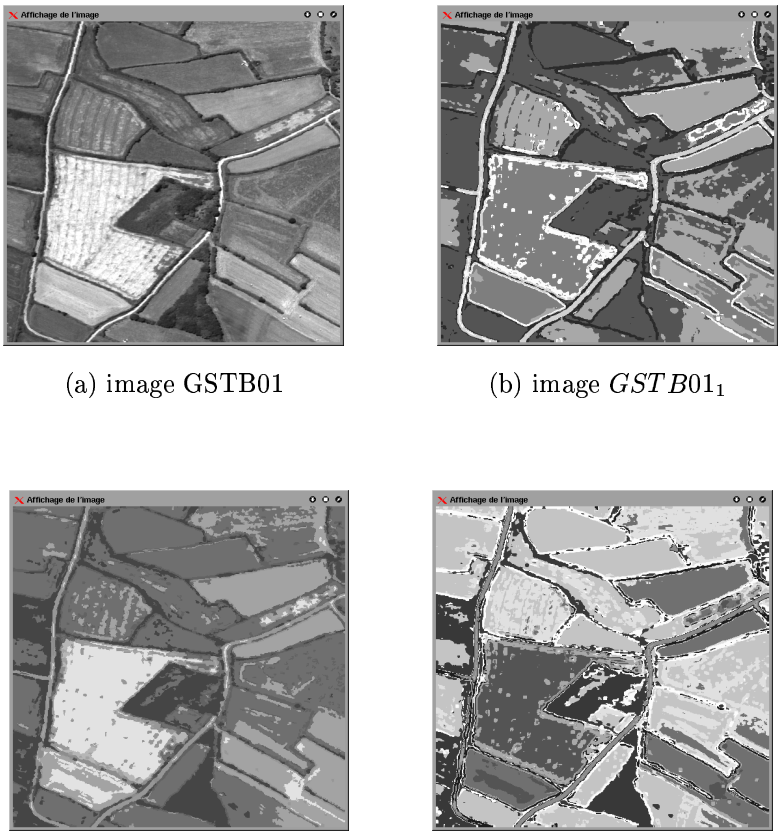
(d) image CAR_3

FIG. 4.4 – *Résultats de segmentation de l'image "CAR"*

d'analyse (3 et 9) et deux valeurs du nombre de classes ($GSTB01_1$ et $GSTB01_3$) (6 et 13). Le résultat $GSTB01_2$ a été obtenu par segmentation adaptative. Les résultats $GSTB01_1$ et $GSTB01_2$ sont assez proches même si le résultat $GSTB01_2$ semble visuellement un peu meilleur. Cette impression est confirmée par la valeur du critère d'évaluation proposé (cf. TAB. 4.3). Le résultat $GSTB01_3$ montre une sur-segmentation de l'image, ce qui se vérifie au vu de la disparité interclasse.

4.3.5 Conclusions

Nous avons proposé un critère d'évaluation d'un résultat de segmentation sans connaissance *a priori*. L'originalité de la méthode développée réside dans sa capacité à évaluer des résultats de segmentation d'images texturées. Nous avons montré la cohérence du critère, d'une part, en le comparant au taux de bonne classification sur des images comportant des régions uniformes et texturées et, d'autre part, par rapport à un jugement visuel.



(a) image GSTB01 (b) image $GSTB01_1$

(c) image $GSTB01_2$ (d) image $GSTB01_3$

FIG. 4.5 – *Résultats de segmentation de l'image "GSTB01"*

Ce critère peut être utilisé pour :

- évaluer plusieurs résultats de segmentation, soit provenant de plusieurs méthodes, soit d'une méthode en modifiant ses paramètres (seuil, attribut,..),
- caractériser un résultat de segmentation dans une étape de fusion.

Nous allons donc l'introduire dans l'algorithme génétique décrit précédemment pour fusionner des résultats de segmentation.

4.4 Méthode de fusion

Après avoir validé le critère d'évaluation d'un résultat de segmentation, nous allons donc l'exploiter dans la méthode de fusion en utilisant un algorithme génétique.

Image	Disparité intraclasse	Disparité interclasse	Critère
$GSTB01_1$	0.34	0.7	0.78
$GSTB01_2$	0.28	0.72	**0.81**
$GSTB01_3$	0.30	0.62	0.77

TAB. 4.3 – *Évaluation des résultats de segmentation de l'image "GSTB01"*

Nous supposons disposer d'un ensemble de n méthodes de segmentation d'image noté $\mathcal{M} = \{M^1,..,M^n\}$. Le résultat de l'application de la méthode M^j sur l'image I est désigné par I^j comportant N^j_{CLASS} classes. On pose $NCOL$ et $NLIG$ le nombre de colonnes et de lignes de l'image I. Dans notre cas, les cinq composantes de l'algorithme génétique sont :

- *génotype* : le résultat de segmentation d'une image I peut être considéré comme un vecteur à valeurs dans $\{1,..,N_{CLASS}\}$ de taille $NCOL \times NLIG$,
- *population initiale* : elle est composée de différents résultats de segmentation sous la forme de vecteurs à valeurs dans $\{1,..,N_{CLASS}\}$,
- *fonction d'aptitude* : elle correspond au critère d'évaluation d'un résultat de segmentation défini précédemment,
- *opérations sur les génotypes* : nous avons utilisé deux des trois opérateurs sur les génotypes décrits précédemment (sélection et reproduction). Nous n'avons pas utilisé l'opération de mutation afin que le résultat final ne dépende que de la population initiale. Le résultat final de fusion ne doit refléter que des informations présentes dans la population initiale. L'opérateur de combinaison de résultats de segmentation consiste à modifier localement un résultat de segmentation en prenant en compte les étiquettes de chaque pixel d'un autre résultat. Si cette modification induit une meilleure valeur du critère, l'individu créé est conservé,
- *critère d'arrêt* : stabilité de l'écart-type des fonctions d'aptitude des résultats de segmentation à une génération donnée.

4.5 Résultats expérimentaux

Nous présentons le résultat de fusion génétique développée de 4 résultats de segmentation de l'image CAR. La méthode de segmentation est une classification des pixels de l'image par un K-means classique à partir de la moyenne et de la variance calculées sur une fenêtre de taille variable (3 et 5) et pour différents nombres de classes (5,10,12,15).

Nous présentons sur la figure FIG. 4.6 les résultats de segmentation de la population initiale et le résultat final de la fusion génétique de l'image “CAR”. Le résulat de fusion des résultats de segmentation obtenu est correct et combine bien

toutes les informations de chaque résultat.

Nous avons fusionné 4 resultats de segmentation d'une image aérienne "GSTB01" (cf. FIG. 4.7) en utlisant la même méthode de segmentation et les mêmes paramètres. Le résultat obtenu est satisfaisant. Nous détaillons dans le tableau TAB. 4.4 plusieurs informations sur la population initiale et finale. La valeur du critère obtenu est nettement supérieure à celles des résultats de segmentation de la population initiale. La convergence est atteinte dans le cas de ces deux images au bout d'une vingtaine de générations.

Information	image CAR	image GSTB01
critère d'évaluation moyen de la population initiale	0.827	0.903
meilleur critère d'évaluation de la population initiale	0.844	0.913
moins bon critère d'évaluation de la population initiale	0.809	0.881
écart type des critères d'évaluation de la population initiale	0.010	0.008
critère d'évaluation moyen de la population finale	0.986	0.979
meilleur critère d'évaluation de la population finale	0.986	0.979
moins bon critère d'évaluation de la population finale	0.986	0.979
écart type des critères d'évaluation de la population finale	5.2e-08	6.02e-08

TAB. 4.4 – *Quelques statistiques du critère d'évaluation des populations initiale et finale des deux images tests*

Nous avons utilisé la méthode de fusion proposée dans une phase de segmentation d'images multi-spectrales. Nous présentons un résultat obtenu sur une image acquise par le capteur CASI de taille 1024×512 pixels avec 10 bandes spectrales (cf. FIG. 4.8). Chaque bande spectrale a été segmentée en utilisant la méthode de multi-seuillage (cf. section 1.3.1.b). Le résultat de segmentation obtenu est satisfaisant visuellement et le nombre de classes détecté est de 11. Ce résultat mériterait d'être validé par une vérité terrain.

4.6 Conclusions

Nous avons proposé une méthode originale de fusion de plusieurs résultats de segmentation en utilisant un algorithme génétique. L'originalité de la méthode réside notamment dans l'introduction d'un critère d'évaluation d'un résultat de segmentation sans connaissance *a priori*.

Nous avons mis en évidence l'efficacité de la méthode au travers de plusieurs résultats expérimentaux.

(a) image originale CAR

(b) résultat de segmentation 1

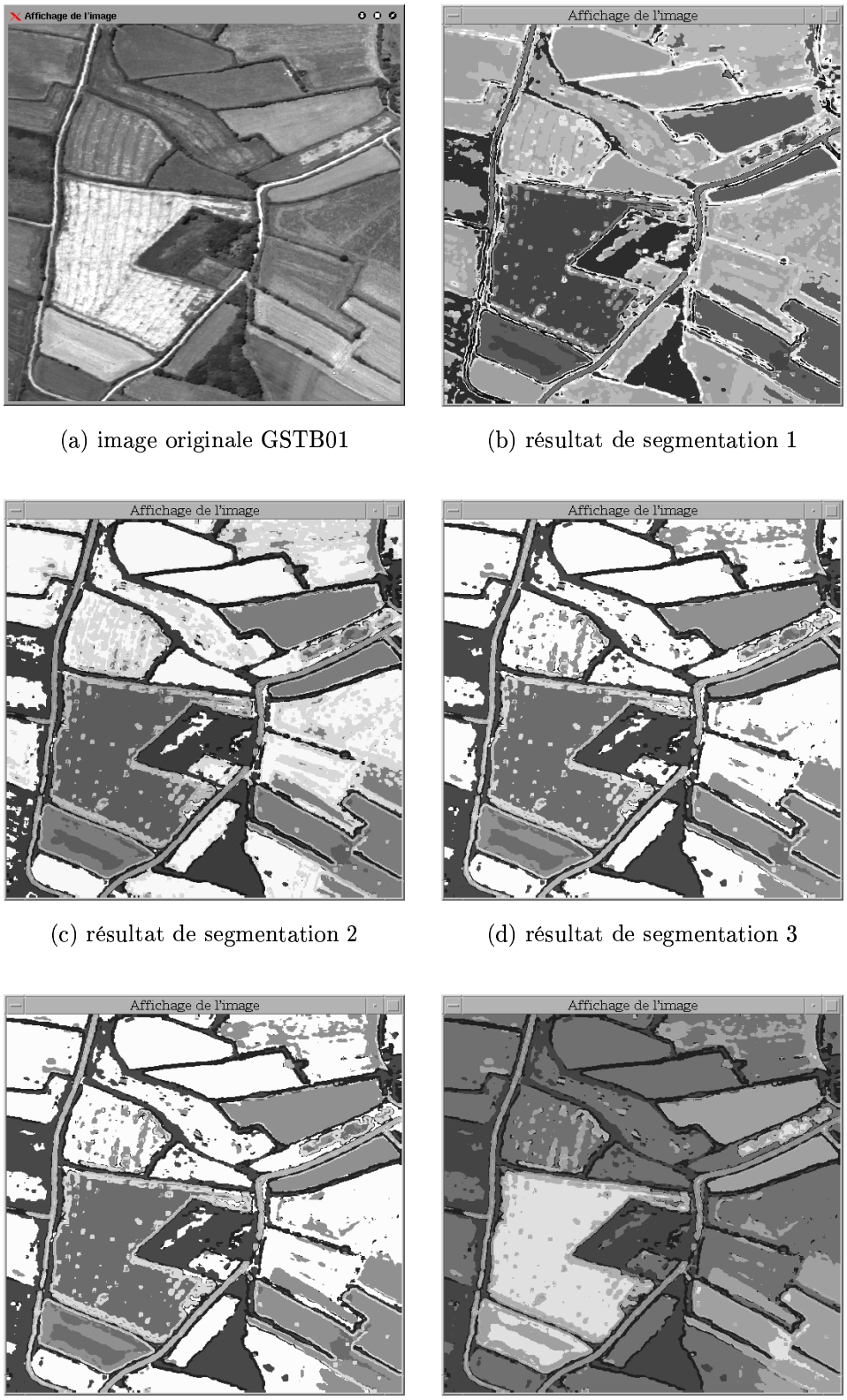
(c) résultat de segmentation 2

(d) résultat de segmentation 3

(e) résultat de segmentation 4

(f) résultat de fusion

FIG. 4.6 – *Fusion de résultats de segmentation de l'image "CAR"*



(a) image originale GSTB01

(b) résultat de segmentation 1

(c) résultat de segmentation 2

(d) résultat de segmentation 3

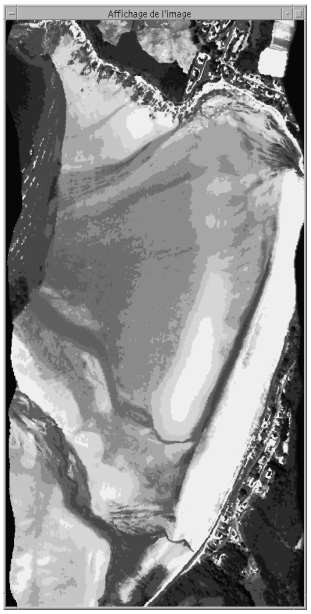
(e) résultat de segmentation 4

(f) résultat de fusion

FIG. 4.7 – *Fusion de résultats de segmentation de l'image "GSTB01"*

(a) bande 1

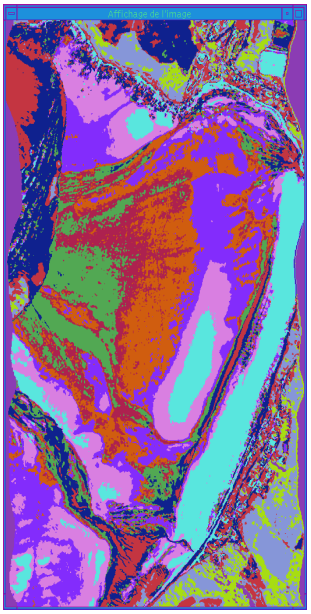
(b) multiseuillage bande 1



(c) multiseuillage bande 6

(d) multiseuillage bande 7

(e) multiseuillage bande 9



(f) résultat de segmentation multi-spectrale

FIG. 4.8 – *Segmentation multi-spectrale de l'image "GSTB001"*

• Propriétés du système de segmentation développé

Nous présentons ici quelques propriétés fondamentales du système que nous avons développé.

Le système proposé a été codé en langage C et sa conception a été réalisée de façon modulaire. Chaque module du système a une tâche spécifique à réaliser. Un module peut se décomposer en plusieurs sous-modules coopérant de façon séquentielle, parallèle ou hybride. La définition des interactions entre sous-modules d'un module est simple d'utilisation. De plus, la conception modulaire du système rend facile son évolution.

Par ailleurs, l'architecture du système proposé permet l'execution en paralléle de plusieurs tâches (la segmentation en parallèle des zones uniformes et texturées d'une image). Afin d'autoriser l'exécution en parallèle des traitements, nous avons géré l'exécution des sous-modules ou processus à l'aide de files de messages. Le principe des files de message est identique à celui des emails. L'exécution d'un module comportant deux sous-modules coopérant de façon séquentielle s'écrit comme suit :

- exécution du premier sous-module du module,
- attendre la fin du sous-module signifié par la réception d'un message de fin de traitement,
- exécution du deuxième sous-module du module,
- attendre la fin du sous-module signifié par la réception d'un message de fin de traitement,
- fin d'exécution du module et envoi d'un message de fin de traitement au système.

Dans le cas d'une coopération parallèle, le module n'attend pas la fin d'exécution du premier sous-module pour déclencher le second. Le gain de temps n'est pas négligeable lorsque les machines utilisées possèdent plus d'un processeur.

Le système développé bénéficie d'une interface conviviale permettant de réaliser des opérations de base en traitement d'images (calcul d'histogramme, détection de contours,..) mais aussi d'appeller les différents programmes développés pendant la thèse (cf FIG. 4.9, FIG. 4.10). Le logiciel sur lequel nous avons implanté notre système de segmentation s'appelle CVIPtools [ZU97].

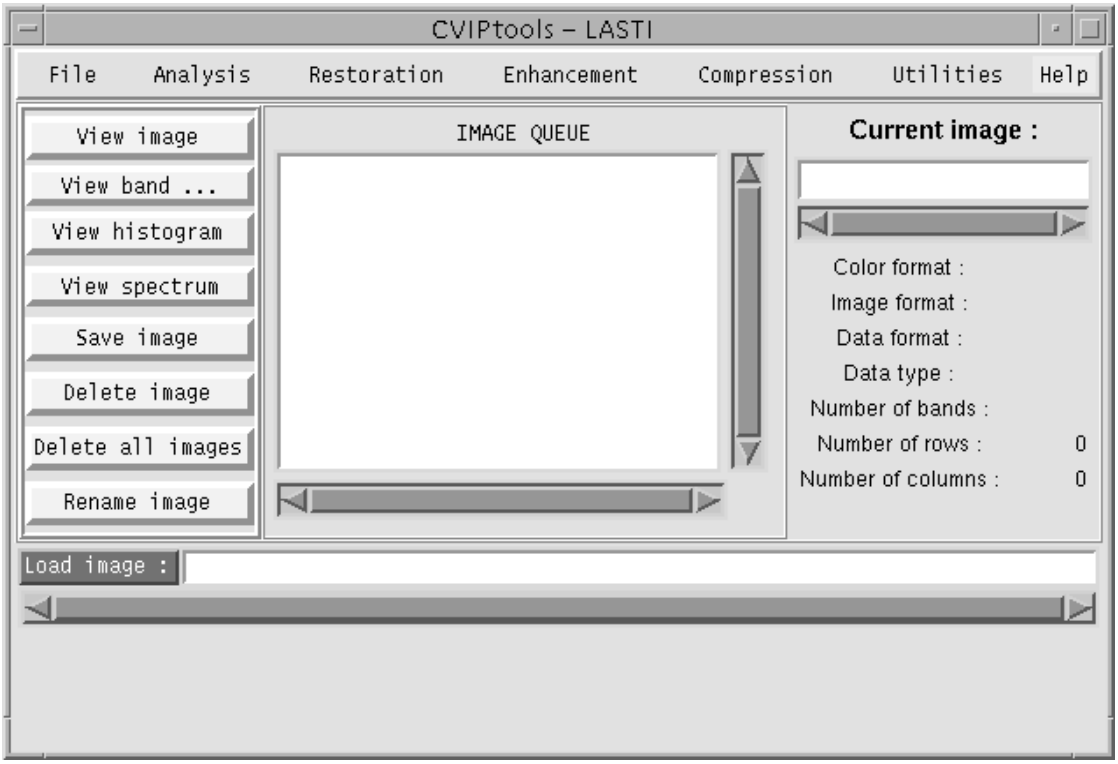


FIG. 4.9 – *Menu principal du système*

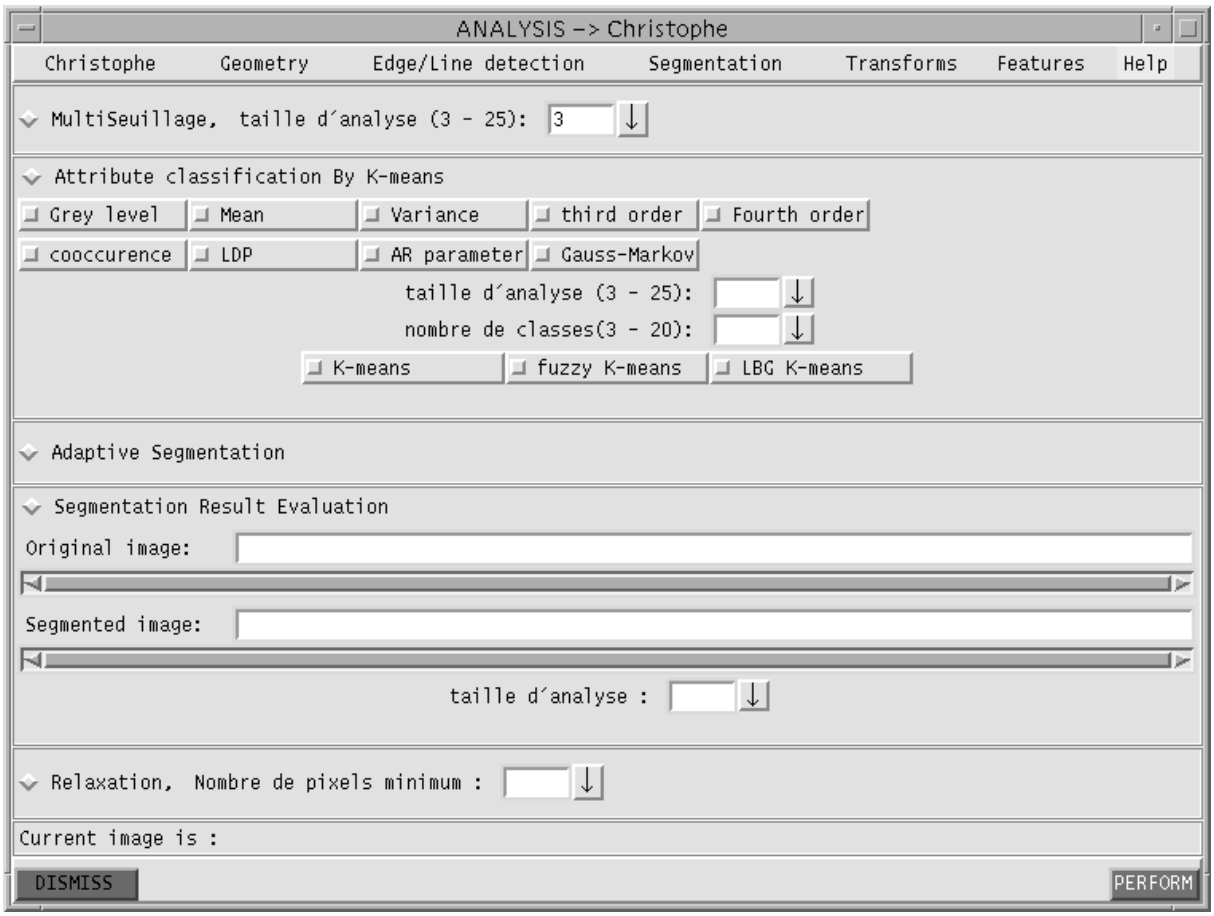


FIG. 4.10 – *Quelques utilitaires en segmentation disponibles dans l'interface*

Conclusions et perspectives

Nous avons proposé dans ce mémoire un système de segmentation original dont la propriété fondamentale est l'adaptation au contexte local de l'image. Chaque module du système a été optimisé afin d'obtenir le meilleur résultat intermédiaire possible :

- Le module d'analyse d'images permet de caractériser finement les différentes régions de l'image à segmenter. Une méthode de détection des zones uniformes et texturées de l'image a été tout d'abord proposée. Pour caractériser les régions uniformes, nous avons utilisé deux attributs locaux (moyenne et variance). Après avoir précisé la notion de motif de texture, nous avons décrits les régions texturées en utilisant deux types de paramètres. Nous avons utilisé des attributs de texture classiques dont nous avons analysé la redondance afin de réduire l'espace de mesures. Nous avons proposé de nouveaux paramètres qui ont été déterminés à partir d'un modèle de texture afin d'avoir une meilleure description de la texture (taille du motif, caractère aléatoire ou déterministe,..). L'ensemble des paramètres retenus a conduit à un taux de bonne classification de 99% sur une base de 300 textures de la base de Brodatz,
- le module de segmentation adaptative utilise une méthode de classification non supervisée que nous avons développée en utilisant les attributs de texture appropriés à chaque type de région,
- le troisième module permet de traiter des images multi-composantes en fusionnant par une approche originale le résultat de segmentation de chaque bande. Cette méthode de fusion basée sur une approche génétique, utilise un critère d'évaluation quantifiant la qualité de chaque résultat de segmentation sans la connaissance d'une vérité de terrain.

Les résultats expérimentaux portant sur différents types d'images ont montré l'efficacité du système développé. L'architecture du système a été conçue afin de permettre l'éxécution en parallèle des différents traitements afin d'optimiser le temps d'éxécution. Une interface a été de plus réalisée afin de faciliter l'utilisation du système ou chacun de ses modules.

Plusieurs améliorations du système sont néanmoins envisageables.

Nous avons montré dans cette étude qu'il est possible d'adapter une méthode de segmentation au contexte local. Pour rendre le système plus robuste, il est souhaitable d'intégrer d'autres méthodes de segmentation (coopération contour-région pour les zones uniformes, méthodes de coopération parallèle pour les régions

texturées, etc..).

Les perspectives de cette étude sont nombreuses :

L'idée directrice de cette thèse a été d'adapter la segmentation au contexte local de l'image. Les résultats de cette approche peuvent être intégrés dans les problèmes de traitement d'images notamment dans les phases de filtrage et restauration aveugles pour faciliter l'estimation des paramètres des dégradations.

La phase de détection des zones texturées et uniformes d'une image peut être aussi utilisée en compression d'images afin d'adapter la description d'une zone de l'image.

ANNEXES

Annexe A

Descripteurs de texture

Afin de caractériser les textures, nous avons retenu pour notre étude les attributs suivants :

- attributs de cooccurrence [HSD73] ;
- attributs de longueur de plages [Lia94];
- attributs des histogrammes locaux [RBR85];
- attributs des extrema locaux [HCR92];
- attributs de l'intégrale curviligne [HCR92];
- la fonction d'autocorrélation.

Des études comparatives non exhaustives ont démontré leur efficacité et leur capacité de sélection [Car93] [RB95]. Le principal désavantage de ces méthodes réside dans le temps de calcul requis pour des images de grande taille et ayant un nombre important de niveaux de gris ainsi que dans la difficulté de choix du vecteur de déplacement.

A.1 Attributs de la matrice de cooccurrence

La méthode de matrice de cooccurrence, développée par Haralick et al [HSD73] consiste à estimer les fréquences d'apparition de couples de niveaux de gris de pixels dans une zone de l'image. Plus précisément, n éléments de la matrice indique la fréquence d'occurrence d'un couple de niveaux de gris x_u et x_v séparés par une distance d et situés sur une droite faisant un angle θ par rapport à l'axe horizontal. La matrice de cooccurrence pour une orientation θ et une distance d est définie par la relation suivante :

$$P_{d,\theta}(g_u,g_v) = card\{(x_r,x_s) \in S, \mid (x_r,x_s) = (x_u,x_v), \|s-r\|_2^2 = d^2, \widehat{(\vec{sr},\vec{i})} = \theta\} \quad \text{(A.1)}$$

où $\vec{i}$ est l'axe horizontal.

Nous détaillons les attributs calculés à partir de la matrice de cooccurrence P pour une distance d et une orientation θ données avec un nombre de niveaux de gris

NG (le terme $P_{d,\theta}(i,j)$ traduit la fréquence normalisée d'apparition d'un niveau de gris i et j distant de d pour une orientation θ. Les différents attributs de la matrice de cooccurrence sont :

1. second moment angulaire : ce paramètre mesure l'homogénéité de l'image ; il est d'autant plus grand que l'image a des transitions de niveaux de gris dominantes.
$$ASM = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} P(i,j)^2$$
2. contraste : il mesure le contraste d'une image ; il est élevé lorsque l'on passe d'un pixel très clair à un pixel très foncé ou inversement.
$$CONTRAST = \sum_{n=0}^{NG-1} [n^2 \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} P_{d,\theta}(i,j) 1_{\{|i-j|=n\}}]$$
3. corrélation : il décrit la corrélation des niveaux de gris d'un pixel avec un autre distant de d dans la direction θ.
$$CORR = \frac{\sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} ij P_{d,\theta}(i,j) - (\sum_{i=0}^{NG-1} iPx(i))(\sum_{j=0}^{NG-1} jPy(j)}{\sqrt{|(\mathcal{X})||(\mathcal{Y})|}}$$
où $Px(i) = \sum_{j=0}^{NG-1} P_{d,\theta}(i,j)$ et $Py(j) = \sum_{i=0}^{NG-1} P_{d,\theta}(i,j)$
et $\mathcal{X} = \sum_{i=0}^{NG-1} i^2 Px(i) - (\sum_{i=0}^{NG-1} iPx(i))^2$ et $\mathcal{Y} = \sum_{j=0}^{NG-1} j^2 Px(j) - (\sum_{j=0}^{NG-1} jPx(j))^2$
4. variance : cet indicateur décrit la dispersion des transitions entre niveaux de gris.
$$VAR = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} [i + 1 - \sum_{k=0}^{NG-1} \sum_{l=0}^{NG-1} kP_{d,\theta}(k,l)]^2 P_{d,\theta}(i,j)$$
5. moment des différences inverses : ce paramètre renseigne sur l'importance des transitions entre niveaux de gris proches.
$$IDM = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} \frac{P_{d,\theta}(i,j)}{1 + (i-j)^2}$$
6. moyennes des sommes :
$$SA = \sum_{k=2}^{2NG} kPxPy(k)$$
où $PxPy(k) = \sum \sum_{i,j,i+j+2=k} P_{d,\theta}(i,j)$
7. variances des sommes :
$$SVAR = \sum_{k=2}^{2NG} (k - SA)^2 PxPy(k)$$

8. entropie des sommes :

$$SENTROPY = -\sum_{k=2}^{2NG} PxPy(k)log(PxPy(k)+\epsilon)$$

où ϵ est une valeur faible (10^{-4})

9. entropie : elle fournit une indication sur le désordre que peut présenter une texture.

$$ENTROPY = -\sum_{i=0}^{NG-1}\sum_{j=0}^{NG-1} P_{d,\theta}(i,j)log(P_{d,\theta}(i,j)+\epsilon)$$

10. variance des différences :

$$DVAR = \frac{NG^2\sum_{i=0}^{NG-1} PXY(i)^2 - (\sum_{i=0}^{NG-1} PXY(i))^2}{NG^4}$$

où $PXY(k) = \sum\sum_{i,j,|i-j|=k} P_{d,\theta}(i,j)$

11. entropie de la différence :

$$DENTROPY = \sum_{i=0}^{NG-1} PXY(i)log(PXY(i)+\epsilon)$$

12. première information sur la mesure de corrélation :

$$ICORR1 = \frac{Hxy - Hxy'}{max(Hx,Hy)}$$

où

$$\begin{cases} Hxy = -\sum_{i=0}^{NG-1}\sum_{j=0}^{NG-1} P_{d,\theta}(i,j)log(P_{d,\theta}(i,j)+\epsilon) \\ Hxy' = -\sum_{i=0}^{NG-1}\sum_{j=0}^{NG-1} P_{d,\theta}(i,j)log(Px(i)+Py(i)+\epsilon) \\ Hx = -\sum_{i=0}^{NG-1} Px(i)log(Px(i)+\epsilon) \\ Hy = -\sum_{i=0}^{NG-1} Py(i)log(Py(i)+\epsilon) \\ Px(i) = \sum_{j=0}^{NG-1} P_{d,\theta}(i,j) \\ Py(j) = \sum_{i=0}^{NG-1} P_{d,\theta}(i,j) \end{cases}$$

13. deuxième information sur la mesure de corrélation :

$$ICORR2 = \sqrt{|1 - e^{-2(Hxy2-Hxy)}|}$$

où $Hxy2 = -\sum_{i=0}^{NG-1}\sum_{j=0}^{NG-1} Px(i)Py(j)log(Px(i)Py(j)+\epsilon)$

14. uniforme : cet indicateur est révélateur de la présence de texture comme nous l'avons montré auparavant pour une distance de déplacement $d = 1$.

$$UNIFORME = \sum_{i=0}^{NG-1} P_{d,\theta}(i,i)$$

15. information contour : ce paramètre fournit une indication sur la présence de contours. Une valeur élevée traduit une transition prépondérante d'un niveau de gris avec les autres.

$$ICONTOUR = max_{i=0,NG-1}(\sum_{j=0}^{i} P_{d,\theta}(i,j))$$

A.2 Attributs de la matrice de longueur de plages

Nous détaillons les attributs calculés à partir de la matrice de longueur de plages PL pour une distance et une orientation données. Galloway [Gal75] a introduit la méthode de matrice de longueur de plages. Elle consiste à rechercher les plages (ou segments) de même niveau de gris. Une plage est définie comme un ensemble consécutif de points ayant des niveaux de gris similaires, dans une direction donnée. Elle est caractérisée par sa longueur et son orientation θ (principalement 0, 45, 90 et 135 degrés). On définit alors les éléments de la matrice de longueurs de plages PL de taille $NGxL$ (où NG est le nombre de niveaux de gris de l'image et L la longueur maximale d'une plage), par :

$$PL(x_s,l,\theta) = card\{(x_{r1},..,x_{rl}) \mid x_{rj} = x_s,\ (\widehat{\overrightarrow{x_{r(j-1)}x_{rj}},\vec{i}}) = \theta,\ \forall j = 1,..,l\}$$

Les attributs de la matrice de longueur de plages utilisés sont les suivants :

1. importance des courtes plages :

$$SRE = \sum_{m=1}^{NG}\sum_{n=1}^{L_{max}} \frac{PL(m,n,\theta)}{n^2}$$

2. importance des longues plages : cet indicateur est d'autant plus élevé que la probabilité de rencontrer des longues plages est élevée.

$$LRE = \sum_{m=1}^{NG}\sum_{n=1}^{L_{max}} PL(m,n,\theta)n^2$$

3. distribution des niveaux de gris : ce paramètre est peu élevé pour une répartition hétérogène des plages pour plusieurs niveaux de gris.

$$GLD = \sum_{m=1}^{NG}(\sum_{n=1}^{L_{max}} PL(m,n,\theta))^2$$

4. distribution des plages : cet attribut mesure la non-uniformité de la répartition des longueurs de plages indépendamment de leurs niveaux de gris.

$$RLD = \sum_{n=1}^{L_{max}}(\sum_{m=1}^{NG} PL(m,n,\theta))^2$$

5. pourcentage de plages : ce paramètre mesure le nombre de plages par unité de surface, il est d'autant plus faible que les plages sont longues.

$$RLP = \sum_{m=1}^{NG} \sum_{n=1}^{L_{max}} \frac{PL(m,n,\theta)}{n_t}$$

où n_t nombre de pixels de la fenêtre d'analyse.

A.3 Attributs des histogrammes locaux du premier ordre

Cette méthode consiste à extraire l'information caractéristique de l'image à partir d'histogrammes locaux du premier ordre [HCR92].

Deux attributs sont obtenus en considérant les histogrammes locaux d'une image sur une fenêtre de taille $N \times N$.

1. module moyen :

$$MOD = \frac{\sum_{i=1}^{NH} ||H_i||}{NH}$$

où $||H_i|| = \sum_{j=1}^{NG}(NG - \frac{n^2}{NG})log(r_i)$, r_i est le nombre de points dans la fenêtre de niveau de gris i et NH est le nombre d'histogrammes calculé sur l'image.

2. phase moyenne :

$$PHASE == \frac{\sum_{i=1}^{NH} phase(H_i)}{NH}$$

où $phase(H_i) = \frac{2\pi j}{NG}$ avec $j = arg\ max_{i=1,NG}(r_i)$

A.4 Attributs des extrema locaux

Cette méthode permet d'évaluer la fréquence relative des pics de luminance. Elle consiste à compter le nombre d'extrema locaux dans une direction θ sur une distance d. L'algorithme se décompose en deux étapes :

- la première phase correspond à un lissage de l'image afin d'éliminer les pics de luminance d'amplitude faible. On pose I_k le niveau de gris au site k, $\tilde{I}_k$ la valeur lissée et S le seuil choisi de détection des extrema. Le calcul de l'image lissée est réalisé de la façon suivante avec $I_1 = \tilde{I}_1$:

$$\begin{cases} si\ \tilde{I}_k \leq I_{k+1} - S/2 & alors\ \tilde{I}_{k+1} = I_{k+1} - S/2 \\ si\ I_{k+1} - S/2 < \tilde{I}_k \leq I_{k+1} + S/2 & alors\ \tilde{I}_{k+1} = I_{k+1} \\ si\ I_{k+1} + S/2 < \tilde{I}_k & alors\ \tilde{I}_{k+1} = I_{k+1} + S/2 \end{cases}$$

  On réalise ce lissage horizontalement et verticalement (l'ordre n'est pas important).

– La seconde étape calcule le nombre moyen d'extrema pour une image, étant données plusieurs directions θ (dans notre cas $\theta = k.\frac{pi}{4}$, $k = 0{,}3$), une distance d'exploration donnée d et le seuil S.

On obtient un vecteur composé de 4 attributs correspondant au nombre d'extrema moyens dans chacune des directions.

A.5 Attributs de l'intégrale curviligne

Le principe de la méthode repose sur l'intégrale curviligne du signal de luminance le long d'une ligne de balayage centrée sur le pixel à traiter ayant une longueur et une orientation données. Le calcul de l'intégrale curviligne le long de la ligne L_i est réalisé de la façon suivante :

$$\int_{L_i} \sqrt{\lambda_1 dx^2 + \lambda_2^2 dy^2 + dI^2_{(x,y)}}$$

où $\lambda_1 = \frac{NLIG}{NG}$,$\lambda_2 = \frac{NCOL}{NG}$ tiennent compte de la résolution ($NLIG$, $NCOL$ sont les nombres de lignes et de colonnes de l'image et NG le nombre de niveaux de gris).

On calcule ainsi un vecteur d'attributs contenant l'intégrale curviligne moyenne dans 4 directions ($\theta = k.\frac{\pi}{4}$, $k = 0{,}3$) pour une distance de balayage donnée d.

Cette caractéristique intègre bien la distribution spatiale des niveaux de gris dans le voisinage du pixel et correspond bien à une mesure locale d'une texture.

A.6 Moments statistiques

Les moments statistiques d'ordre k d'une image I calculés sur une fenêtre W sont définis comme suit :

$$m_k = \sum_{s \in W} s^k.I(s)$$

Les attributs courament utilisés sont ceux correspondant aux ordres 1 à 4.

Bibliographie

[ABHP92] O. Allagnat, J.M. Boucher, D. He, and W. Pieczynski. Hidden markov fields and unsupervised segmentation of images. In *International Conference on Pattern Recognition*, volume 3, pages 96–100, The Hague, August 1992.

[BAJ88] R. BAJCSY. Active perception. In *Proceedings of IEEE*, pages 996–1005, August 1988.

[BB92] J. Benois and D. Barba. Image segmentation by region-contour cooperation for image coding. In *IEEE International conference of Pattern recognition*, pages 331–334, 1992.

[Bel98] F. Bellet. *Une approche incrémentale base de processus coopératifs et adaptatifs pour la segmentation d'images en niveaux de gris.* PhD thesis, Institut Nationnal Polytechnique de Grenoble, June 1998.

[Bes74] J. Besag. Spatial interaction and statistical analysis of lattice system. *Stat. Soc.*, 36:192–226, 1974.

[Bez81] J. C. Bezdek. *Pattern recognition with fuzzy objective function algorithms.* Plenum Press, 1981.

[BG91] O. Beaujard and C. Garbay. Kiss : Un système de vision multi-agents. *RFIA*, pages 89–98, November 1991.

[BHDP95] P. Bonnin, B. Hoeltzener-Douarin, and E. Pissaloux. A new way of image data fusion: the multi-spectral cooperative segmentation. In *IEEE International Conference on Image Processing*, volume 3, pages 572–575, WASHINGTON, October 1995.

[BKS98] N. Benamrane, K. Kies, and J. Shen. Image segmentation by a multiresolution approach. In *EUSIPCO*, 1998.

[Blo96] I. Bloch. Information combination operators for data fusion : a comparative review with classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 1996.

[Bon91] P. Bonnin. *Méthode systématique de conception et réalisation d'applications en vision par ordinateur.* PhD thesis, , December 1991.

[Bro56] P. Brodatz. *Textures : A Photographic Album for Artists and Designers.* Dover, New-York, 1956.

[BS92] C. Bouman and M. Shapiro. Multispectral image segmentation using a multiscale model. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 565–568, SAN FRANCISCO, March 1992.

[BS94] C. A. Bouman and M. Shapiro. Multiscale random field model for bayesian image segmentation. *IEEE trans. IMg. proc.*, 3(2):162–177, March 1994.

[Buf89] J.M.H. Du Buf. Towards unsupervised texture segmentation using gabor spectral decomposition. In *Conference on Image Analysis and Processing*, pages 65–72, POSITANO, September 1989.

[Bur84] P.J. Burt. The pyramid as a structure for efficient computation, 1984.

[BW91] A. Bhalerao and R. Wilson. Multiresolution image segmentation combining region and boundary information. In *Scandinavian Conference on Image Analysis*, volume 2, pages 1162–1169, Aalborg, August 1991.

[CA93] C. C. Chu and J. K. Aggarwal. The integration of image segmentation maps using region and edge information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1241–1252, 1993.

[Can86] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[Car93] J.M. Carstensen. Cooccurrence feature performance in texture classification. In *Scandinavian Conference on Image Analysis*, volume 1, pages 831–838, TROMS, May 1993.

[CC96] C. Cariou and K. Chehdi. Unsupervised texture segmentation using 2-D AR modeling and a stochastic version of the EM procedure. In *EUPSICO*, pages 1877–1880, September 1996.

[Chi94] S. Chiu. Fuzzy model identification based on cluster estimation. *Journal of intelligent and fuzzy systems*, 2(3), September 1994.

[CJA96] E. Chiarello, L.M. Jolion, and C. Amoros. Regions growing with the stochastic pyramid. *Pattern Recognition*, 1:61–75, 1996.

[CK93] T. Chang and C.C Jay Kuo. Texture analysis and classification with tree-structured walevet transform. *IEEE Transactions on Image Processing*, 2(4):429–441, October 1993.

[CL94] Y. Chang and X. Li. Adaptive image region growing. *IEEE Transactions on Image Processing*, 3(6):868–872, November 1994.

[CP95] J. P. Cocquerez and S. Philipp. *Analyse d'images: filtrage et segmentation.* Masson, 1995.

[CR99] K. Chehdi and C. Rosenberger. Analyse et segmentation de textures à partir d'attributs invariants par rotation. In *GRETSI*, 1999.

[CTBP95] C. Collet, P. Thourel, P. Bouthemy, and P. Perez. Détection d'ombres sur images sonar par une analyse multi-résolution et une modélisation markovienne. In *GRESTI*, volume 2, pages 569–572, JUAN-LES-PINS, September 1995.

[CZS89] S. Castan, J. Zhao, and J. Shen. Une famille de détecteurs de contours basée sur filtre exponentiel optimal. In *AFCET-RFIA*, 1989.

[Der87] R. Deriche. Using canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2):167–187, 1987.

[DKZB99] O. Dupuis, V. Kaftandjian, Y. Zu, and D. Babot. Détection de défauts par fusion de signaux ultrasonores et d'images radiographiques. In *GRETSI*, 1999.

[DLPT82] E. Diday, J. Lemaire, J. Poujet, and F. Testu. *Elément d'analyse de données.* Dunod, 1982.

[DLR77] A. Dempster, N. Laird, and D. Rubin. Maximum likehood from incomplete data via the em algorithm. *Journal of royal stastical society*, 1977.

[fB95] R. Kara falah and Ph. Bolon. Une technique d'intégration des résultats en segmentation d'images. In *GRESTI*, volume 2, pages 573–576, JUAN-LES-PINS, September 1995.

[FBC94] R.K. Falah, Ph. Bolon, and J.P. Cocquerez. A region-region and region-edge co-operative approach of image segmentation. In *IEEE International Conference on Image Processing*, volume 3, pages 470–474, AUSTIN, November 1994.

[FJ91] F. Farrokhinia and A.K. Jain. A multi channel filtering approach to texture segmentation. In *Computer Vision and Pattern Recognition*, pages 364–370, LAHAINA, MAUI, HAWAII, June 1991.

[FMP93] J.M. Francos, A.Z. Meiri, and B. Porat. A unified texture model based on a 2-d wold-like decomposition. *IEEE Transactions on Signal Processing*, 41:2665–2678, August 1993.

[Gag83] A. Gagalowicz. *Vers un modèle de textures.* PhD thesis, Université Pierre et Marie CURIE, PARIS VI, May 1983.

[Gal75] M.M. Galloway. Texture analysis using grey level run lenghts. In *Graphical Models and Image Processing*, volume 4, pages 172–179, 1975.

[GG84] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *PAMI*, 1984.

[Har79] R.M. Haralick. Statistical and structural approaches to texture. In *proceedings of the IEEE*, volume 69, pages 786–804, May 1979.

[HB94] J.F. Haddon and J.F. Boyce. Texture classification of segmented regions of flir images using neural networks. In *IEEE International Conference on Image Processing*, volume 3, pages 660–664, AUSTIN, November 1994.

[HCR92] V. Haese-Coat and J. Ronsin. évaluation et comparaison de méthodes de segmentation en imagerie spot. Technical report, INSA RENNES, 1992.

[HH97] W. Hansen and W. E. Higgins. Relaxation methods for supervised image segmentation. *IEEE trans. Pat. An. Ma. In.*, 19(9):949–962, September 1997.

[HJK95a] C. R. Houck, J. A. Joines, and M. G. Kay. A comparison of genetic algorithms, random restart and two-opt switching for solving large location-allocation problem. *Computers, operations research forthcoming in special issue on evolution computation*, 1995.

[HJK95b] C. R. Houck, J. A. Joines, and M. G. Kay. A genetic algorithm for function optimization : a matlab implementation. Technical Report 95-09, NCSU-IE, 1995.

[HP98] F. Huet and S. Philipp. A multi-scale fuzzy classification by knn : Application to the interpretation of aerial images. In *RFIA '98*, volume 1, pages 169–178, Clermont-Ferrand, January 1998.

[HS89] J.Y. Hsiao and A. A. Sawchuk. Supervised textured image segmentation using feature smoothing and probabilistic relaxation techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1279–1292, December 1989.

[HSD73] R.M. Haralick, K. Shanmungan, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, November 1973.

[JS92] S. Castan J. Shen. An optimal linear operator for step edge detection. *CVGIP*, 54:112–133, 1992.

[Jul75] B. Julesz. Experiments in the visual perception of texture. *Scientific software*, 232:2–11, 1975.

[KC98] C. D. Kermad and K. Chehdi. Segmentation d'images : recherche d'une mise en oeuvre automatique par coopération de méthodes. *Traitement du Signal*, 1998.

[KCC95] C. Kermad, K. Chehdi, and C. Cariou. Segmentation d'images par multi-seuillage et fusion de régions labellisées minimisant un critère de similarité. In *Quinzième Colloque GRETSI*, volume 2, pages 641–644, JUAN-LES-PINS, September 1995.

[Ker97] C.D. Kermad. *Segmentation d'image: recherche d'une mise en œuvre automatique par coopération de méthodes.* PhD thesis, Université de Rennes I, July 1997.

[Kir71] R. Kirsh. Computer determination of the constituent structure of biological images. *Computer biomdical research*, 1971.

[KWT88] M. Kass, A. Witkin, and D Terzopoulos. Snakes : active contour models. *Computer vision, Graphics and Image processing*, pages 321–331, 1988.

[KZB92] Z. Kato, J. Zerubia, and M. Berthod. Satellite image classification using a modified metropolis dynamics. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 573–576, SAN FRANCISCO, March 1992.

[Laf96] J. Laferte. *Contribution a l'analyse d'images par modèles markoviens sur des graphes hiérarchiques. Application à la fusion de données multi-résolution.* PhD thesis, université de Rennes 1, October 1996.

[LF93] A. Laine and J. Fan. Texture classification by walevet packet signatures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(15):1186–1191, January 1993.

[Lia94] Q. Liao. *Détection de contours et segmentation d'images : Application à la télédétection et à la biologie marine.* PhD thesis, Université de Rennes I, 1994.

[LMMZ92] D.A. Langan, K.J Molnar, J.W. Modestino, and J. Zhang. Use of the mean-field approximation in an em-based approach to unsupervised stochastic model-based image segmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 57–60, SAN FRANCISCO, March 1992.

[LN85] M. Levine and D. Nazif. Dynamic measurement of computer generated images segmentations. *PAMI*, 7(2):155–164, March 1985.

[LP85] I. Lerman and P. Peter. Elaboration et logiciel d'un indice de similarité entre objets d'un type quelconque. Technical report, IRISA, 1985.

[LPL93] I. Lerman, P. Peter, and H. Leredde. Principes et calculs de la méthode implantée dans le programme chavl (classifiation hiérarchique par analyse de la vraisemblance des liens. *La revue Modulab*, (12):33–70, 1993.

[LS96] Y. Li and F. Santosa. A computational algorithm for minimizing total variation in image restoration. *IEEE Tr. IP*, 5(6):987–995, june 1996.

[MCG92] F. Marques, J. Cunillera, and A. Gasull. Hierarchical segmentation using compound gauss-markov random fields. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 53–56, SAN FRANCISCO, March 1992.

[MH80] D. Marr and E. Hildreth. Theory of edge detection. In *Royal society of london*, 1980.

[MLT95] F. Meriau, O. Laligant, and F. Truchetet. études comparatives de différents détecteurs de contours et segmentation au sens contours par frames multi-échelles. In *GRESTI*, volume 2, pages 617–620, JUAN-LES-PINS, September 1995.

[MRR+53] N. Metropolis, A. W. Rosenbluth, M.N. Rosnbluth, A.H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 1953.

[PD99] N. Paragios and R. Deriche. Geodesic active regions for supervised texture segmentation. In *International Conference on Computer Vision*, volume 1, 1999.

[Pic96] R. W. Picard. A society of models for video and image libraries. *IBM systems journal*, 3, 1996.

[RB95] M.J. Rendas and J.M. Bruneau. Image segmentation by multidimensional anisotropic diffusion. In *GRESTI*, volume 2, pages 653–656, JUAN-LES-PINS, September 1995.

[RBR85] J. Ronsin, D. Barba, and S. Raboisson. Comparaison between cooccurrence matrices, local histogramm and curvilinear integration for texture characterization. In *SPIE*, volume 596, pages 98–111, 1985.

[RC91] A. Rangarajan and R. Chellappa. Image estimation and segmentation using a continuation method. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 2697–2700, TORONTO, May 1991.

[RC97] C. Rosenberger and K. Chehdi. Rapport région 1ère année : Segmentation d'images par méthodes coopératives : application à la télédétection. Technical report, ENSSAT, February 1997.

[RC98] C. Rosenberger and K. Chehdi. Rapport Région 2 ème année : Segmentation d'images par méthodes coopératives : application à la télédétection. Technical report, ENSSAT, September 1998.

[RC99] C. Rosenberger and K. Chehdi. Rapport Région 3 ème année : Segmentation d'images par méthodes coopératives : application à la télédétection. Technical report, ENSSAT, September 1999.

[RCCO99] C. Rosenberger, K. Chehdi, C. Cariou, and J. M. Ogier. Texture analysis of an image by using a rotation invariant model. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, PHOENIX, 1999.

[Saa96] A. Saad. *Filtrage et segmentation d'images radar polarimétriques.* PhD thesis, Université de Nantes, September 1996.

[SGC95] C. Spinu, C. Garbay, and J.M. Chassery. Une approche coopérative et adaptative pour la segmentation d'images. In *GRESTI*, volume 2, pages 609–612, JUAN-LES-PINS, September 1995.

[Sob78] Sobel. Neighbourood coding of binary images for fast contour following and general binary array processing. *CVGIP*, 8:127–135, 1978.

[SP95] F. Salzenstein and W. Pieczynski. Choix automatique de la meilleure méthode de segmentation statistique non supervisée d'images. In *GRESTI*, volume 2, pages 625–628, JUAN-LES-PINS, September 1995.

[Uns84] M. Unser. *Description statistique de la texture.* PhD thesis, EPFL, Lausanne, 1984.

[Uns95] M. Unser. Texture classification and segmentation using walevet frames. *IEEE Transactions on Image Processing*, 4(11):1549–1560, November 1995.

[Wal96] P. Wall. *A Genetic Algorithm for Resource-Constrained Scheduling.* PhD thesis, MIT, 1996.

[WC96] C. S. Won and Y. Choe. Image block classification using stochastic image segmentation. *Electronics Letters*, 32(16):1462–1463, August 1996.

[WM87] B. Wrobel and O. Monga. Segmentation d'images naturelles : Coopération entre un détecteur contour et un détecteur région. In *GRETSI*, 1987.

[WZX92] T. Wang, X. Zhuang, and X. Xing. The use of gibbs random fields for image segmentation. In *ICPR*, volume 3, pages 57–60, The Hague, August 1992.

[Yen86] Li Han Wang Yenping. A new model of motion blurred images and estimation of its parameter. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2447–2449, 1986.

[You89] L. Younes. Parameters estimation for imperfectly observed gibbsian fields. *Prob. Theory and Rel. Fields*, 1989.

[Zad65] L.A. Zadeh. Fuzzy sets. *Infi and Control*, 8:338–353, 1965.

[Zha96a] T. Zhang. Issues in texture classification. CS328B, May 1996.

[Zha96b] Y.J. Zhang. A survey on evaluation methods for image segmentation. In *Computer Vision and Pattern Recognition*, volume 29, pages 1335–1346, 1996.

[ZU97] M. Zuke and S.E. Umbaugh. *CVIPtools: A Software Package for Computer Imaging Education.* Number 3 in 1. Wiley and Sons, 1997.

Liste des tableaux

1.1 Avantages et inconvènients des méthodes de segmentation adaptées aux images uniformes 35
1.2 Avantages et inconvénients de deux types de méthodes adaptées à la segmentation d'images texturées 36

2.1 Valeurs de U et U pour chaque image test et pour différents nombres de niveaux de gris de l'image multi-seuillée 58
2.2 Corrélation des différents paramètres de textures 73
2.3 Corrélation moyenne de chaque type de descripteurs de texture 75
2.4 Nombre d'attributs faiblement corrélés calculé à partir de la matrice de corrélation pour une valeur du seuil de corrélation fixé 80
2.5 Matrice de confusion en utilisant l'ensemble des attributs de texture : taux de bonne classification 75.8% 83
2.6 Matrice de confusion après réduction du nombre d'attributs : taux de bonne classification 76.6% 84
2.7 Paramètres du modèle pour les différentes textures tests 99
2.8 Paramètres du modèle pour les différentes textures tests (suite) . . . 100
2.9 Corrélation des différents paramètres du modèle de textures 102
2.10 Matrice de confusion en utilisant les paramètres du modèle : taux de bonne classification 92.86% 106
2.11 Matrice de confusion en utilisant les attributs de texture et les paramètres du modele : taux de bonne classification 93.48% 107
2.12 Résultats de classification en utilisant différents types d'attributs . . . 109

4.1 Critère d'évaluation et taux de bonne classification des résultats de segmentation de la figure FIG. 3.17 137
4.2 Évaluation des résultats de segmentation de l'image "CAR" 137
4.3 Évaluation des résultats de segmentation de l'image "GSTB01" . . . 140
4.4 Quelques statistiques du critère d'évaluation des populations initiale et finale des deux images tests 141

Table des figures

1.1 Arbre de classification correspondant à la hiérarchie suivante $H = \{\emptyset, a, b, c, d, e, f, ab, abc, de, def, abcdef\}$ 34
1.2 Schéma du système adaptatif de segmentation 51

2.1 Exemple de deux images ayant le même écart-type, l'une texturée (texture damier)et l'autre composée de 2 régions uniformes de taille importante 54
2.2 Transition entre mêmes niveaux de gris dans un voisinage 8-connexe . 55
2.3 Résultats du Multiseuillage des images 56
2.4 Images utilisées pour valider le critère d'uniformité (taille 256×256) 59
2.5 Détection des zones texturées et uniformes de l'image à segmenter . . 60
2.6 Détection des zones uniformes de l'image TELE1 : les zones noires correspondent aux régions texturées 63
2.7 Détection des zones uniformes de l'image TELE2 : les zones noires correspondent aux régions texturées 64
2.8 Détection des zones uniformes de l'image SYNT1 : les zones noires correspondent aux régions texturées 65
2.9 Détection des zones uniformes de l'image SYNT2 : les zones noires correspondent aux régions texturées 66
2.10 Détection des zones uniformes de l'image EXT1 : les zones noires correspondent aux régions texturées 67
2.11 Détection des zones uniformes de l'image EXT2 : les zones noires correspondent aux régions texturées 68
2.12 Mosaïque de textures 72
2.13 Apport de chaque valeur propre pour chaque type d'attribut en pourcentage classé dans l'ordre décroissant 77
2.14 Classification des attributs de texture classiques 79
2.15 Résultat de classement des textures de la mosaïque à partir de l'ensemble des descripteurs de texture avant et après sélection des attributs non corrélés 81
2.16 Présentation des différentes textures de l'album de Brodatz 86
2.17 Interpolation d'un point sur le demi-cercle $\hat{C}_r$ 92
2.18 Détermination de la finesse d'une texture : cas non déterministe . . . 95
2.19 Textures de Brodatz tests 97

2.20 Résultat de modélisation des Textures tests 98
2.21 2 textures pour 4 différentes orientations 101
2.22 Résultats de modélisation des textures à différentes orientations de la figure FIG. 2.21 101
2.23 Classification de tous les attributs de texture 103
2.24 Résultat de classement des textures de la mosaïque à partir des descripteurs de texture et des paramètres du modèle proposé 104
2.25 20 Textures de Brodatz 108

3.1 Schéma des différentes étapes de la méthode de classification proposée 113
3.2 Exemple de l'application de l'algorithme dans le cas de deux classes . 114
3.3 Partitions erronées d'une classe 115
3.4 Résultat de classification d'un nuage de points composé d'une classe et 2 classes 117
3.5 Résultat de classification d'un nuage de points composé de 6 classes par l'algorithme proposé et l'algorithme de nuées dynamiques floues . 119
3.6 Description d'un pixel de l'image sur une fenêtre d'analyse 120
3.7 Segmentation adaptative des images SYNT1, SYNT2 et EXT1 122
3.8 Segmentation adaptative des images EXT2, TELE1 et TELE2 123
3.9 Résultats de segmentation de quelques images acquises par le C.A.S.I 124
3.10 Résultats de segmentation de quelques images acquises par le C.A.S.I. 125

4.1 Découpage en blocs de l'image à traiter 132
4.2 Définition de la disparité intraclasse 133
4.3 10 résultats de segmentation de 5 images tests 136
4.4 Résultats de segmentation de l'image "CAR" 138
4.5 Résultats de segmentation de l'image "GSTB01" 139
4.6 Fusion de résultats de segmentation de l'image "CAR" 142
4.7 Fusion de résultats de segmentation de l'image "GSTB01" 143
4.8 Segmentation multi-spectrale de l'image "GSTB001" 144
4.9 Menu principal du système 146
4.10 Quelques utilitaires en segmentation disponibles dans l'interface . . . 147